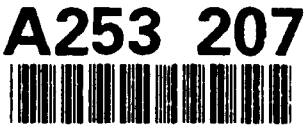
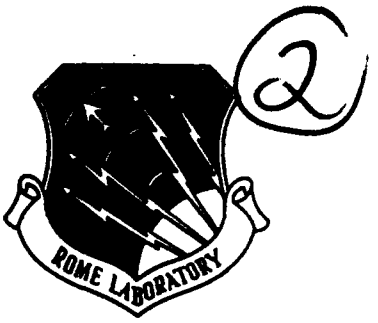


RL-TR-91-429
Final Technical Report
December 1991



VLSI DESIGN FOR RELIABILITY

University of Illinois

Ibrahim N. Hajj, Vasant B. Rao, Ping Yang, Richard Burch,
Hungse Cha, Russell Iimura, Farid N. Najm

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

Rome Laboratory
Air Force Systems Command
Griffiss Air Force Base, NY 13441-5700

This report has been reviewed by the Rome Laboratory Public Affairs Office (PA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

RL-TR-91-429 has been reviewed and is approved for publication.

APPROVED:

Mark T. Pronobis

MARK T. PRONOBIS
Project Engineer

FOR THE COMMANDER:

John J. Bart

JOHN J. BART
Technical Director
Reliability & Compatibility Directorate

If your address has changed or if you wish to be removed from the Rome Laboratory mailing list, or if the addressee is no longer employed by your organization, please notify RL(ERD) Griffiss AFB, NY 13441-5700. This will assist us in maintaining a current mailing list.

Do not return copies of this report unless contractual obligations or notices on a specific document require that it be returned.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave Blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	December 1991	Final Jan 89 - Jan 90

4. TITLE AND SUBTITLE

VLSI DESIGN FOR RELIABILITY

5. FUNDING NUMBERS

C - F30602-88-D-0028
Task N-9-5716
PE - 62702F
PR - 2338
TA - 01 WU - PX

6. AUTHOR(S)

Ibrahim Hajj, Farid N Najm, Ping Yang

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Coordinated Science Lab
University of Illinois
1101 W Springfield Ave
Urbana IL 61801

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Rome Laboratory
Griffiss AFB NY 13441-5700

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

RL-TR-91-429

11. SUPPLEMENTARY NOTES

RL Project Engineer: Mark T Pronobis/ERDD/(315) 330-4758
Prime Contractor: University of Dayton, Dayton OH 45433

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for Public Release; distribution Unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

This report describes the work accomplished during the period Jan 89 to Jan 90 on reliability analysis of VLSI CMOS circuits for electromigration prevention. The work involved three subtasks: (1) development of probabilistic simulation techniques to replace expensive exhaustive circuit simulation, (2) extraction and modeling of transistor circuit description and parasitics from layout, and (3) extraction of metal bus and line models from layout, and calculation of the current densities.

In probabilistic simulation, new techniques for computing expected and variance current and voltage waveforms in CMOS circuits are developed and implemented in the program CREST. A proof that electromigration median-time-to-failure predictions are mathematically related to expected and variance current density waveforms in the busses is also given. For very large designs, transistor level probabilistic simulation, although significantly faster than exhaustive deterministic circuit simulation, is still not cost-effective. Thus the development of probabilistic macromodeling simulation techniques is initiated. Initial results show that such an approach is feasible. This will make it possible to include to include other technologies, such as BiCMOS.

14. SUBJECT TERMS

Reliability; probabilistic simulation; electromigration; VLSI CMOS circuits; macromodeling; extraction

15. NUMBER OF PAGES

64

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

NSN 7540-01-280-5500

Standard Form 298 (Rev 2-89)
Prescribed by ANSI Std. Z39-18
298-102

Block 13 Cont (Abstract)

In the area of layout extraction, a "new" circuit extractor has been developed and linked to the Oct/Vem design data base system. The extractor is being used to extract a test transistor artwork circuit specified in CIF. The output of the extractor supplied the inputs to both CREST and a bus modeling and analysis program. It provides the link between the simulation and the physical design.

In the work on power and ground bus modeling and analysis, computationally efficient techniques for extracting the bus model by identifying basic shapes of metal regions and reprenting them by circuit models are developed. These circuit models are precomputed using finite element numerical methods. Techniques for computing current density waveforms in the bus given the current density waveforms at the bus contact (these are provided by CREST) are also developed. These current density waveforms will be used in the estimation of electromigration effects in the bus, which will then be linked to the layout through Oct/Vem.

EVALUATION

This technical report describes the development of computer-aided design techniques for the reliability analysis of VLSI CMOS circuits for electromigration susceptibility. The work involved three subtasks: (1) development of probabilistic simulation techniques for determining the expected current waveforms, (2) circuit extraction from the physical layout, and (3) extraction of the power busses and calculation of current densities. While this work forms the basis for a electromigration analysis tool for VLSI CMOS circuits, additional research needs to be accomplished in all three subtasks mentioned above.

Additional information on the calculation of the expected and variance current waveforms can be found in the complementary report "VLSI DESIGN FOR RELIABILITY", Sep 89 to Nov 89.

Mark T. Pronobis

MARK T. PRONOBIS
Project Engineer

VLSI Design for Reliability

Final Report

Account No. 93928-04/P.O. 61575X

Contract No. F30602-88-D-0028 (Task N-9-5716)

I. Introduction

This task is concerned with the development of computer-aided design techniques, as well as, software tools for reliability analysis and design of Application Specific Very Large Scale Integrated (ASIC VLSI) CMOS circuits. There are many failure mechanisms that have to be considered when analyzing the reliability of integrated circuits. These include metal and contact electromigration, power supply voltage drop and noise, transistor degradation induced by hot-electron injection into the oxide, oxide breakdown, electrostatic discharge through input protection circuit, system-noise induced latchup, and charge-particle-induced soft errors.

Since wearout mechanisms leading to poor reliability are a result the circuit behavior over typically long periods of time, they depend *on the combined effects* of a large number of possible input waveforms rather than worst-case conditions. One of the main issues addressed in this task is the development of efficient techniques for representing the combined effects of all possible input waveforms by a one or possibly two waveforms (expected and variance waveforms), and the derivation and implementation of fast techniques for generating these waveforms. This has led to the development of a new methodology which we call probabilistic simulation [1-4].

In this task, we concentrated our efforts on a specific reliability problem, namely, electromigration (EM) and, showed how our probabilistic simulation techniques could be applied

to deal with this reliability problem. The methodologies and the computer-aided design tools that we develop, however, will also lay the foundation for addressing the other reliability issues mentioned above.

In addition to the probabilistic simulation techniques, which are explained in the next section, our work effort has included the development and implementation of extraction techniques and software which are used in extracting transistor netlist and parasitics from CIF layout files. Extraction is an essential step for both simulation and design; it forms the link between circuit simulation and the physical layout of the circuit. Our work on extraction will be described in Section 3. In Section 4, we describe our work on the extraction and modeling of the power and ground busses from the CIF layout file. Techniques for computing the current densities in the form of expected and variance waveforms at various points in the bus in terms of probabilistic current waveforms applied at contact points in the bus are also described. In Section 5, we summarize our results and discuss the work that still needs to be done.

II. Probabilistic Simulation Techniques

A detailed report explaining the probabilistic simulation method applied to CMOS circuits based on transistor netlist description of the design is attached, and thus will not be repeated here. The method has been implemented in computer program CREST [1]. Even though the method described in the report is much faster than exhaustive circuit simulation such as SPICE, it is still not fast enough to handle very large designs since it is based on a flattened transistor circuit specification.

During the last four months of the contract period, we embarked on the development of a macromodeling probabilistic simulation approach to augment the one used in CREST. In our opinion, this approach will be suitable for simulating very large designs described in an hierarchical fashion. In the following, we summarize the results we obtained so far on this macromodeling approach.

We consider each macromodel to be a "black box" which can represent a simple logical gate or a higher level macro, such as a flip flop. The macromodeling capability is needed for four reasons. First, it greatly increases the flexibility of CREST and allows quick expansion of current estimation to new technologies. Second, the input circuit description is in the form of a netlist which is available at early stages of the design and takes up less room than a transistor-level description. Third, high-level macromodels have the promise of increased execution speed if the problems associated with them can be solved. Finally, the user can define macromodels that utilize his knowledge of the circuit to simplify the simulation. For instance, a flip-flop defined at the transistor level currently is classified as a supergate and is relatively costly for CREST to simulate; however, it could easily be replaced by a single macromodel that would be much easier to simulate.

Each macromodel is implemented as a set of separate C routines that are linked into the core of CREST. This allows faster simulation than parameterized models would. In this object-oriented programming approach, each macro data structure contains pointers to the C routines that handle its interconnection and simulation. Therefore, no lengthy searches are required during simulation.

With this approach the changes to the core routines of CREST are minimized. CREST expects the simulation routine of each macro to accept transitions on its input nodes, calculate and schedule appropriate current pulses, and schedule transitions on the output nodes. Any routines that fulfill these interface requirements can be used as macromodels. Therefore, a great deal of flexibility exists in creating models. Since the routines for each macromodel can be quite lengthy, a macromodel generation program has been written to generate these routines from a description of the macromodel. This generator takes a file that describes the macro and creates the necessary routines. To create this file, two approaches are possible. First, a menu-driven user interface has been developed that allows the user to specify a macro with all the features described below. This interface is extremely useful for prototype modeling and will continue to be useful for allowing a user to examine and, if necessary, modify a macromodel in his library.

The goal of any CAD tool is to simplify and improve the design process without adding extra burdens on the designer. Since much of the information needed for macromodeling already exists in the data base, the macromodel generator should make use of this information. The suggested flow chart is shown in Figure 2.1. Since many different data formats are used in industry, certain translators would have to be written for each user.

The prototype version of the macromodel generator allows for basic macromodels [Fig. 2.2], and it will be expanded as examples are found that require additional features. The job required of the macromodeler can be divided into four areas: physical description, functional description, timing specification, and current estimation. The physical description of a macro is used to connect the macro in the circuit. Functional descriptions are required for each output node of the macro. This specifies the relationship between the signal and transition probabilities of the input and output nodes. The conversion of deterministic specification (logical values) of functionality to probabilistic specifications is the most complex problem in macromodeling. Timing specifications are necessary to determine the delay between any transition on the input nodes and a corresponding transition on the output nodes; while, current estimation information determines the amount of current that should be scheduled given transitions on the inputs and outputs of the macros.

The physical description of the macro is the simplest problem in macromodeling. For each macromodel, a simple list of nodes is provided which specifies the node name (used during functional description), the type of the node (INPUT, OUTPUT, POWER, or GROUND), and the capacitance from each node to Vcc and ground. This information is used in a very straightforward manner to determine how to hook up each macro and how much capacitance each macro contributes to the interconnecting network of the circuit.

Two types of timing specifications are supported. The basic timing information is represented as a switching time for the output node given that an input switches. The formula supported is:

$$Ts = A + B*Cext$$

Where A and B are constants and Cext is the capacitance external to the output node. Switching equations are specified for a low to high (Tslh) and high to low transition (Tshl) on each output node. Since most probabilistic events have probability of a low to high transition (Plh) and another probability of a high to low transition (Phl), an estimated transition time is used for the event on the output node.

$$Ts = (Tslh*Plh + Tshl*Phl)/(Plh+Phl)$$

For large macros, this simplistic approach is not sufficient. In many cases the delay from input to output is not the same for all inputs. This is approximated by delays on each input so that the sum of the delay plus the estimated Ts is equal for all input to output pairs. This is implemented in CREST by creating an extra global node and scheduling an event on the node after the appropriate pin to pin delay (Figure 2.3).

Two types of current estimation are used in the macromodeling version of CREST. The simplest type is DC current. For each input or output node of the macro, a high dc current and low dc current can be specified. The total dc current at any time is the sum over all input and output nodes of the probability low times the low dc current plus the probability high times the high dc current. Implementing DC current required changes in CREST, since CREST previously only kept track of piecewise-linear transient sources. CREST now keeps track of two source types: piecewise-linear transient sources and piecewise-constant dc sources. The two types of sources are merged together during output to produce the total source current. Transient current pulses are supported by the macro modeler. Currently only triangular pulses are allowed; however, CREST is capable of handling any type of piecewise-

linear pulse. As more knowledge is gained from examples, additional pulse types will be implemented. The width of each triangular pulse is estimated from timing information, and the area of the pulse is determined from the probability of transition and the capacitance at the output node (Figure 2.4).

The most complex task in macromodeling is the conversion of deterministic (logical) functional description of the macro into probabilistic functional description. Three probabilities are required for each transition event on an output node: the probability that the node was high before the event, the probability of a transition from low to high, and the probability that the node will be high after the event. Three types of logical functions are supported by the macromodeler: combinational (boolean) functions, static memory functions, and dynamic memory functions. The conditions under which a boolean output node is high can be expressed as a simple boolean equation consisting only of the present values of the input nodes. Conversion of a boolean equation to a set of probability equations is the basic problem and conversion of memory functions are an extension. Static memory outputs depend on the current and previous states of input and output nodes. Furthermore, this output node always has a conducting path to either ground or Vcc. Dynamic memory outputs also depend on current and previous states of input and output nodes. However, the node may not have a direct path to either ground or Vcc; the output state may be held by capacitor. These macros must be handled differently, since the output state can be overridden by changes in other macros that share the same global output node. Figure 2.5 shows examples of macros with all three types of output functions.

The approach implemented for conversion of boolean functions to probabilistic equations is based upon work done by Parker and McClusky [5]. This work gives an algorithm for

transforming boolean equations to signal probability equations by repeatingly applying the following identities:

$$Ph(A) = Ph(A)$$
$$Ph(\sim A) = 1 - Ph(A)$$
$$Ph(A \wedge B) = Ph(A)*Ph(B)$$
$$Ph(A \vee B) = Ph(A) + Ph(B) - Ph(A)*Ph(B)$$

with the special algebra rule that Ph(A)*Ph(A) = Ph(A). For example,

$$Ph((A \wedge B) \vee (A \wedge C)) = Ph(A)*Ph(B) + Ph(A)*Ph(C) - Ph(A)*Ph(B)*Ph(C).$$

This approach can only be used for static signal probabilities; however, we developed a similar approach that can be used to get transition probabilities. Three signal probabilities are used in this approach: Probability high (Ph), probability low to high transition (Plh), and probability low before transition (Plp). The following identities are then used:

$$Plh(A) = Plh(A)$$
$$Plh(\sim A) = 1 - Ph(A) - Plp(A) + Plh(A)$$
$$Plh(A \wedge B) = Plh(A)*Ph(B) + Plh(B)*Ph(A) - Plh(A)*Plh(B)$$
$$Plh(A \vee B) = Plh(A)*Plp(B) + Plh(B)*Plp(A) - Plh(A)*Plh(B)$$

Three special algebra rules are also used in this transformation.

$$Plp(A)*Plp(A) = Plp(A)$$
$$Ph(A)*Ph(A) = Ph(A)$$
$$Plp(A)*Ph(A) = Plh(A)$$

This approach works; however, the number of terms in the final equation can be exponential. This is especially obvious for Plh, but is also true of Ph. A second approach was developed

and implemented that kept track of eight probabilities, instead of three. These eight probabilities are the probabilities that the transition is low and high before the transition and after the transition (Plp, Php, Pl, Ph), and all four possible transition probabilities (Pll, Plh, Phl, Phh). In the worst case, the new approach is still exponential; however, on many test examples the complexity of the final equations is significantly reduced.

Static memory functions are represented as a state table of the output and input nodes before (t-) and after (t+) of each transition, as in the following example.

Toggle Flip Flop

Out(t+)	Out(t-)	In1(t+)	In1(t-)	In2(t+)	In2(t-)
1	1	0	X	0	X
1	X	1	X	X	X
1	0	0	X	1	0

The output probability is specified as the probability of the union of these states.

$$Ph(Out(t+)) = P(inputs(t+) \wedge inputs(t-) \wedge Out(t-))$$

Which can be broken down to:

$$Ph(Out(t+)) = P(inputs(t+) \wedge inputs(t-)) * P(inputs(t-) \wedge Out(t-))/P(inputs(t-)).$$

P(inputs(t+) ∧ inputs(t-)), and P(inputs(t-)) can be solved using the techniques developed for boolean functions. The first step in solving P(inputs(t-) ∧ Out(t-)), is to analyze the state table to determine if any constraints that can be used to simplify P(inputs(t-) ∧ Out(t-)). These constraints identify any present input conditions that uniquely determine the state of the output, and any redundant states. After this simplification, independence of the output is

assumed.

$$P(inputs(t-) \wedge Out(t-)) = P(inputs'(t-) \wedge Out'(t-))$$
$$= P(inputs'(t-))*P(Out'(t-))$$

This independence assumption is unavoidable since the current value of Out may actually depend on the value of Out a number of transitions ago, and CREST only maintains information about the present transition.

Dynamic memory functions are implemented by dividing the macro into two components (Figure 2.6). The state at the output is determined by a boolean function at node T which goes through a pass transistor network. Thus,

$$Ph(out(t+)) = Ph(T(t+)*Pon(path(t+)) + Ph(out(t-))*Poff(path(t+)).$$

After this simple transformation, all the probabilities can be determined with the approaches developed for boolean equation transformation.

Currently, our macromodeling research is focusing on two different areas. The first area is a continuation of our search for better transformations of boolean equations into probabilistic equations. One of the most promising approaches comes from a literature search through the field of system reliability. One of the problem in this field is to determine the probability that a path exists between two points given a boolean description of the possible paths. This is very similar to our boolean transformation problem, for static signal probabilities. The best results seem to be obtained by generating disjoint boolean forms. If A, B, and C are disjoint, then $Ph(A \vee B \vee C) = Ph(A) + Ph(B) + Ph(C)$. This is a big simplification; however, generating disjoint forms increases the complexity of the boolean expression. Although the worst

case number of boolean expressions is exponential, published data suggests that on most practical examples this approach is relatively efficient. We have derived a way of utilizing the disjoint form for transition probabilities:

$$Plh(A \vee B \vee C) = Plp(A \vee B \vee C)\left(\frac{Plh(A)}{Plp(A)} + \frac{Plh(B)}{Plp(B)} + \frac{Plh(C)}{Plp(C)}\right).$$

With this identity and the other techniques already presented, the disjoint approach is a promising improvement. Another place for improvement involves simplification using a factoring approach; research is just beginning on this approach.

The second area that we are focusing on is automatic macromodel generation. We are in the beginning phases of this and are just now defining the problem and the direction that research should go. If successful, this part of the research could significantly enhance the applicability of macromodeling and the speed of CREST. Part of the work will be to examine the potential of our approach for hierarchical simulation.

III. Layout Extraction

3.1. Introduction

In this section, we summarize our work on layout extraction by considering a test example provided to us by researchers at RADC. The example is a 10,000-transistor artwork specified in a CIF file. We will refer to this test example as the "2uchp chip"

The artwork design process is shown in Figure 3.1. Extraction provides the link between the *design* of artwork and its *verification* through simulation. The Oct/Vem system (to be described later) will be used to provide a design framework in which to carry out the artwork design.

In the following, the extraction problem will be described first. Next, the Oct/Vem system will be presented and its role in our desired design system will be described. The progress of the extraction work will then be presented, and finally the future plans for the extractor will be described.

3.2. Circuit Extraction

Circuit extraction is the transformation of artwork information into circuit information so that one may verify that a circuit performs its intended function. The process of circuit extraction is shown in Figure 3.2. The input to the extractor consists of the mask geometries of the layout. Usually the layout consists solely of rectangles. The geometrical extraction step consists of, first, the identificatior of transistors by finding the regions where POLY and DIFF mask layers overlap. Secondly, all of the interconnection nets are identified by starting at each transistor's drain or source region and collecting all of the electrically-connected rectangles. The extraction of electrical parameters is next, where the dimensions of the

transistors are calculated, and the resistance and capacitance of the interconnection nets are modeled and calculated. The final output consists of the transistors extracted and the parasitic resistances and capacitances of the interconnect.

For typical extractors the input artwork is typically described in *CIF* (Caltech Intermediate Form), an ASCII format where each layout geometry is described in terms of its shape (usually a rectangle), mask layer, and coordinates. The output produced is typically in Spice format which can be read by most circuit simulation programs. An example of CIF input for a given layout and its extracted Spice file is given in Figure 3.3.

CIF is usually produced as the output of a layout design tool but it is **not** meant to be a *design language*--in other words, it is extremely hard to edit a CIF file to change a layout. Its purpose is to provide an intermediate format between design tools. The Oct/Vem system, on the other hand, fully supports design activities.

3.3. The Oct/Vem System

Oct/Vem is a VLSI database and design system developed at UC Berkeley. More specifically, *Oct* is a database system capable of storing the design data for an entire VLSI chip, including the schematic, artwork, and the extracted netlist of a design. The basic unit of storage is a *cell*, which, for example, may contain a single transistor, a logic gate, or entire functional unit (ALU, CPU, or RAM). To efficiently represent the design information, cells are *hierarchical*, so that a cell may contain instances of other cells. In addition, parameters of the design may be annotated to the structural information. This may include the dimensions for a transistor, or parasitic capacitance values for nets (substrate capacitance). *Vem* is the graphical editor for Oct elements; circuit schematics or artwork may be created and edited

with this tool.

Since an Oct database can provide a repository for all design data--artwork as well as an extracted netlist--both artwork and simulation tools can access the database for a design. If all tools read directly from the Oct database, all data conversions would be eliminated between tools that require different file formats (for example CIF, Spice format). Storing all design data in the Oct database will make it easier to make changes to a design as well as verify its function. Therefore, in the long-term it may be advantageous to develop a VLSI design system where layout data is stored in the Oct database and the circuit information is also returned to the database after extraction so that simulators can read this information directly.

3.4. Evaluation of Two Extractors

To develop a suitable extractor, two existing extractor implementations developed at the University of Illinois were studied. The final extractor will use a combination of the better parts of the two as the basis for its design with added enhancements. The new extractor is being integrated into Oct/Vem to extract the 2uchp chip, the extractor will be enhanced to ensure that it can efficiently extract large designs.

Both extractors take CIF-coded rectangles as input and generate a Spice-format file as output. Basically, the two extractors differ in their geometric extraction algorithms.

The first extractor, iCPEX, uses 4-d binary search-trees to store the input rectangles. Each node in the tree corresponds to a rectangle in the layout. At any node in the tree, one of the coordinates of the rectangle is used to bisect the input plane and any subsequent rectangles that are inserted into the tree are placed in the node's right or left sub-tree depending where

the new rectangle lies in relation to the bi-section line. To perform geometric extraction, the basic operation is called *intersection search*, in which all rectangles that intersect a given region are found. Let N be the number of input rectangles. Intersection search is O(log N) for this data structure. Since this must be done for all input rectangles, geometric extraction should take O(N log N) for iCPEX.

The second extractor, Ace, uses a scanline extraction algorithm. Again the basic operation is to find all intersecting rectangles. Rather than considering all of the rectangles at once, the rectangles are only considered when they cross a vertical *scanline* that sweeps across the input plane from left to right. The scanline stops at points in the plane where the rectangles begin and end. There are O(N) of these stopping points. At each stopping point, all rectangles that are *beginning* (ie., the scanline is at leftmost edge of the rectangle) are *inserted* into a "dictionary" data structure; all rectangles that are *ending* are *deleted* from the "dictionary." The "dictionary" structure is a data structure that supports insertion, search, and deletion of rectangles; this may be implemented as a height-balanced tree where insertions and deletions are O(log N). When a rectangle is added to the "dictionary" structure, it is determined if it intersects with any other rectangles currently active in the structure. Therefore the time complexity of geometric extraction for Ace should also be O(N log N).

3.5. Results

The 2uchp chip design is described in a CIF format file. It consists of 74 hierarchical blocks. The layer names used by the two extraction programs did not exactly match those in the original CIF file. As a result, 2uchp's original CIF description was converted into the Oct format; then Oct utility programs were used to output a CIF description again, but with the

proper layer names. A side benefit of reading the CIF into Oct is that the cells can now be viewed or edited in Vem.

After an extensive evaluation period, it was determined that the iCPEX implementation was not adequate for our problem, especially that the geometric extraction in iCPEX is actually $O(N^2)$ in some parts of the code. To find all other rectangles intersecting a given rectangle, since the "other rectangles" were in a linked list (not a balanced tree), the search was O(N). For N rectangles, the operation became $O(N^2)$. As a result, iCPEX was too slow.

On the other hand, the Ace code was easier to read and seemed robust. Since Ace handles CIF descriptions with no instances of sub-cells allowed, i.e., does not allow hierarchical descriptions, new code had to be added to input hierarchical cells. As a comparison with iCPEX, Ace extracted a 500-transistor cell in 50 seconds that originally took iCPEX 35 minutes to accomplish. Therefore, Ace has been adopted as the basis for the geometric extraction portion of the final extractor that is currently being developed.

The new Ace-based extractor is also able to extract the power and ground bus rectangles for electromigration analysis. For example, in Figure 3.4 the original rom2 cell is shown. The new Ace-based extractor then extracts just the power and ground bus rectangles along with the contact points to produce a layout for use by the power and ground bus modeling program, which is described in the next section.

3.6. Future Plans

As designs reach the 100,000 transistor level or more, it becomes important to use memory space efficiently. Exploiting the hierarchy of the design to avoid storing repetitive information will become necessary. This will be a key contribution of the final extractor.

In hierarchical extraction, the user-defined hierarchy is preserved as much as possible. All sub-cells of a given cell must be extracted before the given cell itself is extracted. Hierarchical extraction is not trivial since sub-cells may overlap. Overlapping rectangles can create or delete transistors, or create or break connections between electrical nodes.

To eliminate the need for a CIF file to represent the artwork that is also stored in the Oct format, the extractor will be changed to directly read the Oct database format. As is the case now, the output of circuit primitives will be in Spice format.

We have observed that geometric extraction is O(N log N). It could be much more time-consuming to compute the parasitic capacitances and resistances of the interconnect in addition to the transistor netlist. For instance, to compute the coupling capacitance between each rectangle and every other rectangle in a circuit is an $O(N^2)$ process. Therefore, it will be important to implement the extractor on a shared-memory multiprocessor for higher performance.

IV. Power and Ground Bus Modeling

4.1 Introduction

In this section, we summarize our work on extracting the RC model of the power and ground busses and on computing the current densities in the busses, given the current waveforms at the contacts of the busses. These current waveforms are the ones produced by CREST after performing probabilistic simulation on the extracted transistor circuit.

Ideally, the current density in a conductor is linearly related to the gradient of the potential distribution in it. This distribution may be obtained by solving Laplace's equation, which has mixed (Dirichlet and Neumann) boundary conditions for typical metal regions found in VLSI circuits with complex geometries. Consequently, it is impossible to obtain the potential distribution analytically. The only feasible, accurate, and rigorous approach is to use a numerical technique, such as the finite-element method (FEM) [6]. However, an accurate FEM analysis proves to be computationally very expensive for very complex metal patterns in VLSI circuits such as the power bus and ground bus.

The goal of this research has been to develop a computationally efficient approach to accurately determine the current density at various regions in a metal bus by exploiting certain properties of the geometry of the bus. This approach has been motivated by examining the *equipotential plots* obtained from simulating certain simple geometrical shapes using the FEM. Right-angle bends, abrupt line-width changes, and contact regions are examples of such shapes. Rules derived from these plots enabled the subdivision of a relatively complex conductor pattern along breaklines, which roughly correspond to straight-line equipotential contours. This allowed us to define a set of *primitive* shapes of metal regions. Although the

equipotential lines are not exactly parallel to the boundary of these primitives, they are close enough to being parallel so that this assumption leads to a very good approximation.

These primitive shapes were individually analyzed earlier in a *preprocessing phase* with FEM, and formulas for the equivalent resistance and capacitances for each primitive shape were derived in terms of their geometrical dimensions. The *on-line simulation phase* begins with identifying the various primitive shapes within the given metal bus and constructing an electrical RC network for the overall bus. Given the current loading waveforms at the various contact nodes from CREST, a transient analysis of this linear network is performed yielding the voltage waveforms at the different nodes. With these voltage waveforms, the current waveforms in each resistive branch are computed. Finally, the current waveform in a resistive branch is divided by the width of the corresponding primitive region resulting in the *bulk-current-density* waveform for that region. For certain primitives with right-angle bends, the value of the *peak-current-density* at the corner can be computed at each time-point from the corresponding bulk value by using a certain formula given later in this report.

4.2. Preprocessing using FEM

The electrostatic potential distribution ϕ in any region that is free of sources and sinks satisfies a partial differential equation known as the Laplace's equation. For VLSI metal conductors, whose lateral dimensions far exceed the depth, the variation of ϕ perpendicular to the surface of the conductor can be neglected. Therefore, we can define the potential function $\phi(x, y)$ as depending only on the 2-dimensional surface coordinates x and y, and independent of the depth coordinate z. Consider a 2-dimensional metal region A of arbitrary geometry. Let ∂A denote the boundary of the region A. This boundary can be divided into

two distinct regions, namely, ∂A_ϕ denoting *conducting faces* (or parts of the boundary where the potential function is fixed), and ∂A_ν denoting an *insulating face* (or parts of the boundary connected to some insulator). The Laplace's equation for the potential function ϕ on region A can be written as

$$\nabla^2\phi\,(x, y) = 0$$

with *mixed boundary conditions:*

(a) *Dirichlet* : $\phi(x,y) = f(x,y)$ on the conducting faces ∂A_ϕ of the boundary, where f(x,y) is a known function, and

(b) *Neumann* : $\frac{\partial\phi}{\partial\nu} = 0$ on the insulating faces ∂A_ν of the boundary, where $\frac{\partial\phi}{\partial\nu} = \nu\cdot\nabla\phi$ is the normal component of the gradient $\nabla\phi$ along the boundaries, with ν being the unit normal vector pointing outward at the boundary.

Clearly, the Neumann boundary conditions ensure that no current passes through the insulating faces.

In the FEM approach, the entire metal region is discretized into a collection of simple elements such as rectangles and triangles. The potential function in each element is then assumed to be a simple function such as linear for triangles or bilinear for rectangles which can be determined entirely at any point inside the element from only the potential values at the vertices (or corners) of that element. The individual contributions from each element are then stamped on to a global *stiffness matrix* resulting in a highly sparse linear system of equations for the unknown potentials at the vertices of the elements which is then solved using sparse matrix techniques. The accuracy of an FEM simulation, however, improves as the sizes of the individual elements become smaller which increases the number of elements

needed to cover the entire region.

After numerically solving for the potential ϕ using the FEM on certain geometric shapes, *equipotential* contour plots are obtained by an in-house software package using X-windows and these plots can be displayed on a Micro-VAX workstation. Some examples of such equipotential plots are shown in Figures 4.1 and 4.2. Figure 4.1 shows an right-angle bend, while Figure 4.2 illustrates a metal region corresponding to a line-width variation. From these plots it is clear that the equipotential lines tend to become parallel at a distance which is approximately one line-width from the corner.

Another geometrical shape that was examined is a metal rectangle containing a square contact (or via) placed symmetrically in the middle with respect to the width of the overall rectangle. For the purposes of reducing the computation of the FEM analysis it was also assumed that only one of the two ends of the metal rectangle is a conducting face. An equipotential plot for such a shape is shown in Figure 4.3. Examining this plot shows that the equipotential lines can be assumed to be parallel at distances of approximately one width (of the rectangle) away from the edge of the contact. The condition for parallelism in this plot was that the potentials along a straight line parallel to a conducting face does not vary by more than 1 percent.

It can be easily seen that if any metal region A is broken into two sub-regions A_1 and A_2 such that the common boundary between the two sub-regions is an *equipotential* curve of the original region A, then solving the Laplace's equation on A is equivalent to solving the equation on A_1 and A_2 separately with an additional Dirichlet-type boundary condition introduced at the common boundary in each case. It is this observation that motivates the definition of

primitive regions discussed below.

4.3. Primitive Regions

In this research we consider only those metal regions that can be viewed as a composition of the following *primitive* regions shown in Figures 4.4 to 4.9. In each figure we show the geometrical parameters of the primitive and the corresponding equivalent RC circuit. For each primitive, the darkened faces on the boundary are the conducting faces while the rest of the boundary is an insulating face. In the following discussion, R_{sh} is the sheet resistance of the metal in *ohms per square*, i.e., $\Omega/\square$, and C_{ox} is the capacitance per unit area of the metal in $pF/\mu m^2$, and the unit for all geometrical dimensions is the micron, i.e., μm. Both these values can be obtained from the technology file provided by the user. The primitive regions are :

(S) Straight-line segment of length l and width w as shown in Figure 4.4. The equivalent resistance is $R = R_{sh}\frac{l}{w}$ and capacitance is $C = C_{ox}\, w\, l$.

(L) L-shaped right-angle bend of smaller width w_1 and larger width w_2 as shown in Figure 4.5. The equivalent resistance shown in the figure is $R = R_{sh}(\alpha-2)$, where α is a correction factor for the resistance due to the bend obtained by FEM analysis and is shown by the solid line in Figure 4.10 as a function of w_1/w_2. The equivalent capacitances shown in Figure 4.5 are computed as $C_1 = C_{ox}w_2^2/2$, $C_2 = C_{ox}(w_2^2+w_1w_2)/2$, $C_3 = C_{ox}(w_1^2+w_1w_2)/2$, and $C_4 = C_{ox}w_1^2/2$.

(W) Width-change of smaller width w_1 and larger width w_2 as shown in Figure 4.6. The equivalent resistance is $R = R_{sh}\beta/2$ where β is a correction factor for the resistance due to

the width change obtained by FEM analysis and is shown by the dashed line in Figure 4.10 as a function of w_1/w_2. The equivalent capacitances are $C_1 = C_{ox}w_2^2/2$, $C_2 = C_{ox}(w_1^2+w_2^2)/2$, and $C_3 = C_{ox}w_1^2/2$.

(V) Via or contact region with a square via of size w_c located symmetrically in the middle of a rectangle of width w and length $2w+w_c$ as shown in Figure 4.7. The equivalent resistance is $R = R_{sh}\gamma$, where γ is a correction factor for the resistance due to the contact region obtained by FEM analysis and is shown in Figure 4.11 as a function of w_c/w. The equivalent capacitance is $C = C_{ox}(2w^2+w_cw-w_c^2)/4$.

(T) Three-way junction containing an inner rectangle of size w_1 by w_2 as shown in Figure 4.8. The equivalent resistances are $R_1 = R_{sh}\frac{w_2}{2w_1}$ and $R_2 = R_{sh}(\alpha - 2 - \frac{w_2}{2w_1})$. The correction factor α is computed from the solid line in Figure 4.10 as a function of w_{min}/w_{max} where $w_{min} = \min\{w_1,w_2\}$ and $w_{max} = \max\{w_1,w_2\}$. The equivalent capacitances are $C_1 = C_{ox}w_1^2/2$, $C_2 = C_{ox}w_2^2/2$, and $C_3 = C_{ox}w_1w_2$.

(F) Four-way junction containing an inner rectangle of size w_1 by w_2 as shown in Figure 4.9. The equivalent resistances are $R_1 = R_{sh}\frac{w_2}{2w_1}$ and $R_2 = R_{sh}\frac{w_1}{2w_2}$. There is no correction factor introduced in the resistances in this case. The equivalent capacitances are $C_1 = C_{ox}w_1^2/2$, $C_2 = C_{ox}w_2^2/2$, and $C_3 = C_{ox}w_1w_2$.

We wish to emphasize that contacts or vias cannot be present in any of the primitives L, W, T, or F. This is considered a violation in our approach. Moreover, the FEM analysis of the primitives serves two purposes :

1. It is used to verify the assumption that the equipotential lines are indeed parallel to the conducting face of the primitive if the corresponding limb of the primitive is extended.
2. It is used to calculate the correction factors for the equivalent resistances in case of primitives L, W, and V.

There are some fundamental problems with the FEM analysis due to the finite memory size and limited accuracy of the computer. As mentioned earlier, the FEM analysis becomes more accurate as the number of elements used to discretize the given region is increased. Hence, the FEM analysis for each primitive was performed using different sized discretization elements and the most conservative value of the resistances were stored.

4.4. On-line Simulation

We have implemented a computer program written in the C language that reads in a description of a metal bus in California Intermediate Format (CIF). The CIF file is first read into the MAGIC system and written out again in a *maximally horizontal strip* (MHS) form. This means that whenever there is more than one way of describing the geometry of rectangles, the description that maximizes the horizontal length is taken. For example, consider the overall region shown in Figure 4.12(a). A description of this region as a collection of two rectangles shown in Figure 4.12(b) is not in MHS form. However, the description shown in Figure 4.12(c) is in the MHS form. A routine which transforms an arbitrary CIF description into MHS form will eventually be written in the next phase of this research project.

After reading in the CIF file in MHS form, the program attempts to identify the various primitive regions in the metal bus region. The primary criterion to be satisfied is that two primitives can be abutted only at their conducting faces. Furthermore, these two faces must

have matching dimensions. Since the geometry of the metal bus can be arbitrarily complex, the identification of primitives is quite involved. The program actually executes a number of phases. At each phase of the program, if a shape is encountered that the program cannot handle, the program flags an error, finishes that phase, and then exits, so that the user can modify the layout to conform to the set of primitives.

The program begins by creating a linked list of rectangles. A separate list is maintained for different CIF layers. There is also a separate list just for the labels. These labels will become important when the transient analysis is performed. The program then performs the following operations on each metal layer separately.

Each metal rectangle is identified as being either *horizontal* or *vertical.* A metal strip is horizontal (vertical) if the horizontal (vertical) length is at least twice the vertical (horizontal) length. If it is neither, then it is flagged, and the program exits when it has finished identifying all the other metal rectangles. An error that has been detected is printed out with the information on the routine that detected it and its location. This horizontal/vertical identification indicates the direction of current flow. Current is assumed to flow in the direction of the longer length. This process simplifies the algorithm but presents a problem when there is isolated metal rectangle due a contact which may be neither horizontal nor vertical. In this case, the designer needs to go into the layout and attach an extra strip to the contact so that it now becomes either vertical or horizontal.

Two abutting horizontal rectangles are combined if the combination results in a horizontal rectangle, otherwise an error is flagged at that location. The idea behind this is that we want the current flowing in these rectangles to see the maximum width. Figure 4.13 shows a valid combination, and Figure 4.14 shows an instance where error is flagged. There are 6

different combinations of abutting horizontal rectangles as shown in Figure 4.15. Each of these combinations needs to be taken care of separately because the number of new rectangles that must be created is different and the geometry is also different.

The L, T, and F primitives are formed when a vertical rectangle abuts a horizontal one. Hence, the program looks for this pattern and breaks the rectangles accordingly. Once again, there are 6 combinations to consider and each of these combinations needs to be addressed separately as shown in Figure 4.16. The "center" rectangle is labeled as a "primitive" rectangle for use by a subsequent routine. An error is flagged if the abutting side of one of the rectangles doesn't completely overlap the abutting side of the other rectangle as shown in Figure 4.17.

The program now searches the entire list of rectangles and for each rectangle that is labeled as being "primitive", it breaks the adjacent rectangles to form an L, T, or F, primitive shape as shown in Figure 4.18. The broken rectangles next to the "primitive" rectangles are always squares in accordance with the definitions of these primitives. An error will be flagged if any of the adjacent rectangles are found to have insufficient size as shown in Figure 4.19.

The program then looks for abutting straight-line segments that have different widths to form the W primitives. This situation can occur both vertically or horizontally. If the program encounters such a pair, it breaks the rectangles further so that a square exists right next to the width change as shown in Figure 4.20. Once again, an error is flagged if there is insufficient length to form the squares.

At this stage the identification of all L, W, T, and F primitives is complete. If no errors have been flagged, the program then links the contact vias with the rectangle that it is enclosed in. If the enclosing rectangle is a part of any of the primitives identified thus far, an error is flagged; otherwise, the program breaks the enclosing rectangle as shown in Figures 4.21 and 4.22. The criterion to be satisfied by the enclosing rectangle is that the distance between each of its conducting faces and the closest contact edge must be at least the width of the enclosing rectangle. The situation where both faces of the enclosing rectangle are conducting is shown in Figure 4.21 while the situation when only one face is conducting is shown in Figure 4.22. In either case, the two squares adjacent to the contact via are defined to form a V primitive. Once again, an error is flagged if the enclosing rectangle does not meet the above length requirements.

Any rectangle that is not a part of any of the primitives identified above is defined to be an S primitive. Hence, the order in which the primitives are identified is :

(1) The L, T, and F primitives,

(2) The W primitive,

(3) The V primitive, and finally,

(4) The S primitive.

After the identification of all the primitives is completed and no errors are flagged, the program begins to construct an RC network by combining the equivalent RC circuits for each primitive as shown in Figures 4.4 to 4.9. The program now numbers each node of the constructed RC network as follows. First, the node corresponding to the contact via labeled VDD (for power bus) or GND (for ground bus) is numbered 0. Next, the nodes corresponding to

the other contact vias are assigned a number by a counter which starts at 1 and is incremented each time a node number is assigned. Finally, the remaining nodes are assigned node numbers also from the same counter. The program then outputs a SPICE file containing the description of the constructed linear RC network. The piece-wise linear (PWL) descriptions of the current loading waveforms as given by CREST are added to the SPICE file along with SPICE commands for the printing time-step and time-interval for transient analysis and a list of node numbers at which output waveforms are required.

The circuit simulator iPRIDE [7] is used to perform a complete transient analysis of the above RC network. The voltage waveforms produced by iPRIDE are read back into our program for post-processing as follows. For each resistor in the current waveform through the resistor is computed which is then divided by the cross-sectional area of the corresponding region of the primitive to result in the current density waveform J_{bulk} in the *bulk region* of the primitive. The cross-sectional area referred to here is the product of the width of the region and the thickness of the conductor as given by the technology file. For primitives containing sharp corners such as L, W, T, and F primitives the *peak current density* J_{com} at the corner is computed using a formula derived in [8] as

$$J_{com} = 1.04\, J_{bulk} \left[\frac{1+\delta^{-2}}{\varepsilon} \right]^{1/3}$$

where ε is the ratio of the radius of curvature of the corner to the width of the narrower limb of the primitive and δ is the ratio of the width of the wider limb to that of the narrower limb of the primitive. Given a value of the radius of curvature at the corners which in turn depends on the metal deposition technique and photolithography constraints, the above equation is used to compute the current density magnitude at each corner in the layout. While

Scanning Electron Microscopy (SEM) failure analyses of damage due to electromigration show that the right-angle bend is not a preferential damage site, it might still be useful to know how much larger than the bulk value is the magnitude of the current density is at each corner, since local heating may be caused by this singular point.

V. Summary

In this report, we have described the work we have accomplished during the past year on reliability analysis of VLSI circuits. The work involves three subtasks: (1)development of probabilistic simulation techniques to replace expensive exhaustive circuit simulation, (2) extraction and modeling of transistor circuit description and parasitics from layout, and (3) extraction of metal bus and line models from layout and calculation of the current densities.

In probabilistic simulation, we have developed new techniques for computing expected and variance current and voltage waveforms in CMOS circuits and implemented the techniques in program CREST. We also proved that electromigration median-time-to-failure predictions are mathematically related to expected and variance current density waveforms in the busses. For very large designes, transistor level probabilistic simulation, although significantly faster than exhaustive deterministic circuit simulation, is still not cost-effective. We thus initiated the development of probabilistic macromodeling simulation techniques. This is described in Section 2. Initial results show that such an approach is feasible. This approach will allow us to include other technologies, such as BiCMOS. However, as mentioned at the end of Section 2, much more work is still needed in this area. In particular, we need to develop automatic macromodeling techniques and hierarchical probabilistic methods, which we are planning to do during the next year.

In the area of layout extraction, we have developed a "new" circuit extractor after evaluating two "old" ones and linked the new extractor to the Oct/Vem design data base system. The extractor is being used to extract a test transistor artwork circuit specified in CIF and provided to us by researchers at RADC. The output of the extractor supplies the inputs to both CREST and the bus modeling and analysis program. It provides the link between the

simulation and the physical design. Again, because of the size and the complexity of VLSI circuits, we need to develop hierarchical extraction techniques, which we are in the process of doing.

In the work on power and ground bus modeling and analysis, we have developed computationally efficient techniques for extracting the bus model by identifying basic shapes of metal regions and representing them by circuit models precomputed using finite element numerical methods. We also developed techniques for computing current density waveforms in the bus given the curent density waveforms at the bus contact (these are provided by CREST). We are in the process of developing computationally efficient techniques for analyzing very large bus circuit models, estimation of electromigration effects in the bus, linking the results to the layout through Oct/Vem. The detection of high expected current densities in contacts and their effects will also be studied. In addition, voltage drop estimation in busses and signal lines could be added without much difficulty.

Publications and References

[1] F. Najm, R. Burch, P. Yang, and I N. Hajj, "CREST-A Current Estimator for CMOS Circuits," *IEEE International Conference on Computer-Aided Design (ICCAD-88)*, Santa Clara, CA, pp. 204-207, November 1988.

[2] F. Najm, I. Hajj, and P. Yang, "Electromigration Median Time-to-Failure Based on a Stochastic Current Waveform," *Int. Conf. on Computer Design (ICCD '89)*, Cambridge, MA, pp. 447-450, October 1989.

[3] F. Najm, I. N. Hajj, and P. Yang, "Computation of Bus Current Variance for Reliability Estimation of VLSI Circuits," *Int. Conf. on Computer-Aided Design (ICCAD-89)*, Santa Clara, CA, pp. 202-205, November 1989.

[4] F. Najm, R. Burch, P. Yang, and I. N. Hajj, "Probabilistic Simulation for Reliability Analysis of CMOS VLSI Circuits," *IEEE Transactions on Computer-Aided Design*, Vol. 9, No. 4, pp. 439-450, April 1990.

[5] K. P. Parker and E. J. McCluskey, "Probabilistic Treatment of General Combinational Networks," *EEE Transactions on Computers*, pp. 668-670, June 1975.

[6] P. Tong and J. N. Rossettos, *Finite-Element Method*, Cambridge, MA: MIT Press, 1967.

[7] M. C. Chang and I. N. Hajj, "iPRIDE: A Parallel Integrated Circuit Simulator Using Direct Method," *Proceedings of ICCAD-88*, Santa Clara, CA, pp. 304-307, November 1988.

[8] F. B. Hagedorn and P. M. Hall, "Right-Angle Bends in Thin Strip Conductors," *Journal of Applied Physics*, Vol. 34, No. 1, pp. 128-133, January 1963.

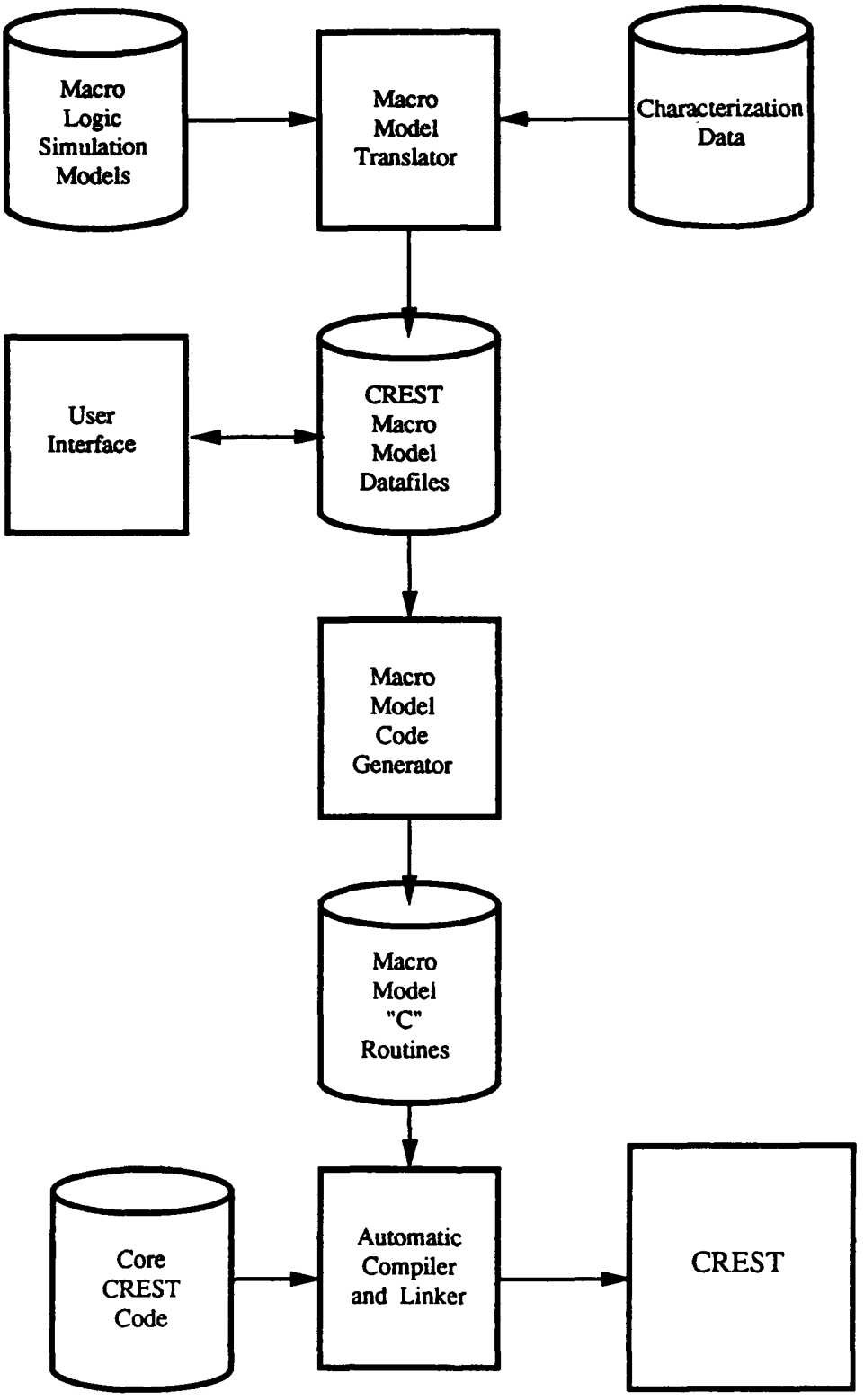


Figure 2.1 - CREST with Macro-models

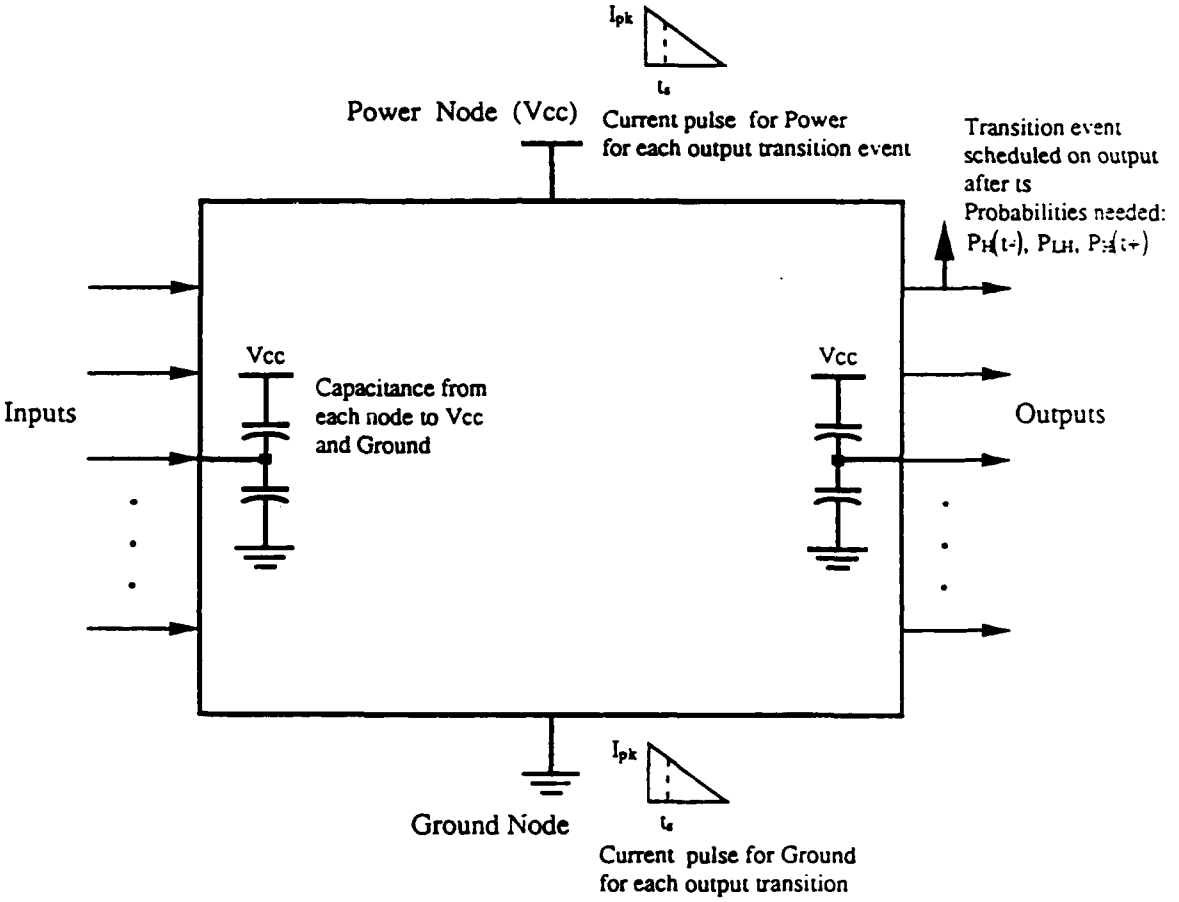


Figure 2.2 - CREST Analysis

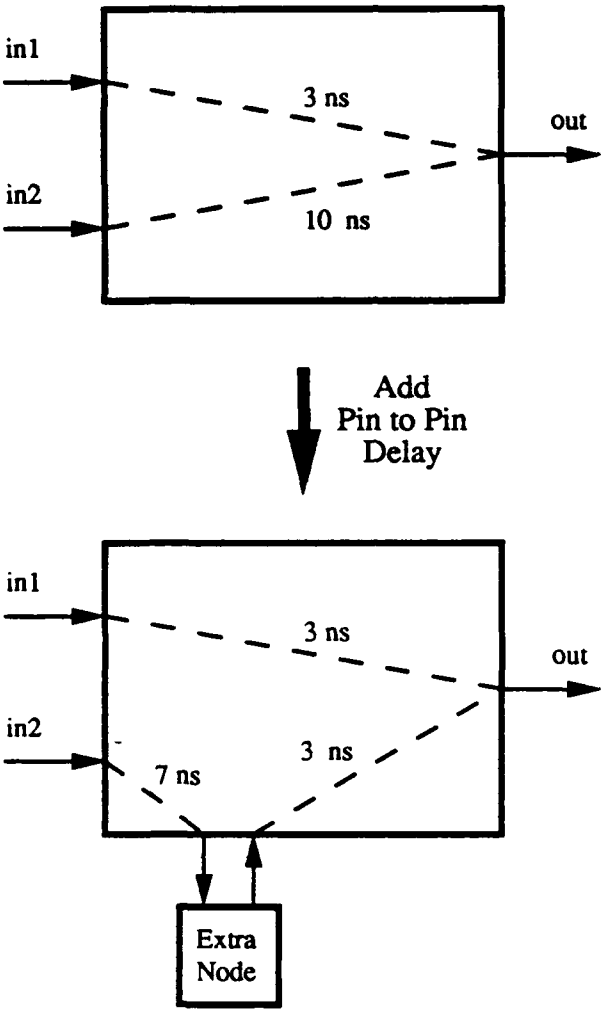
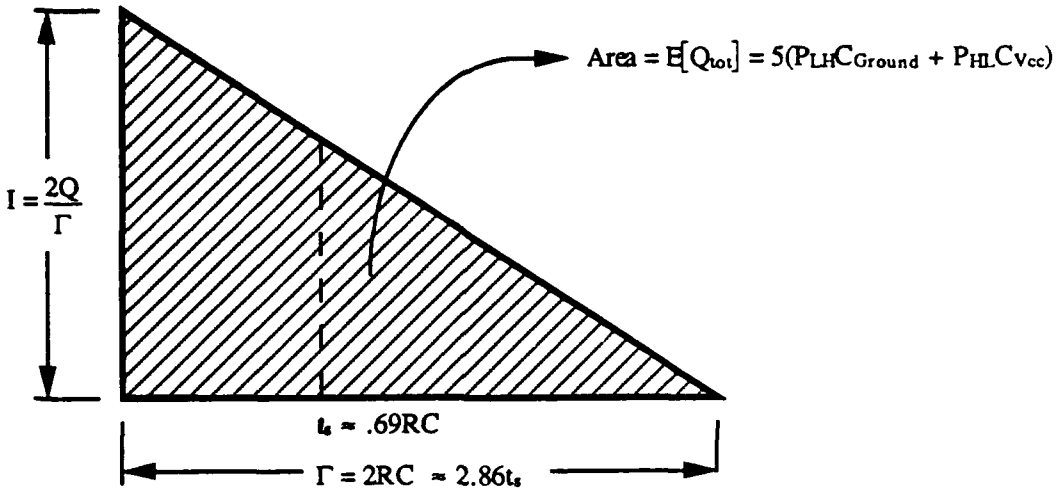
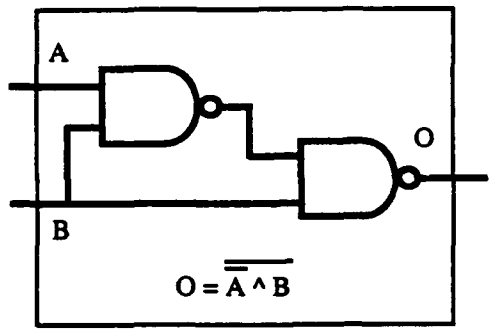


Figure 2.3 - A transition on in2 takes 10 ns to affect out, while a transition on in1 only requires 3ns

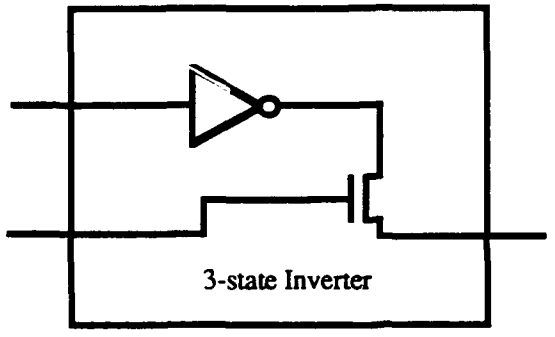


(t_s is frequently a function of capacitance)

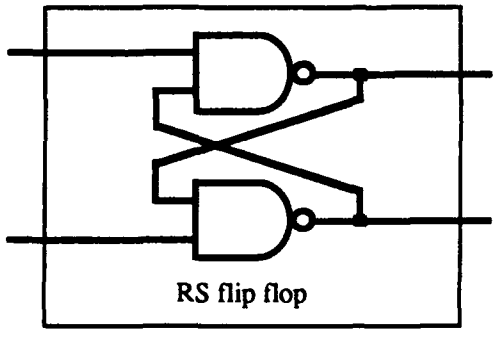
Figure 2.4 - Triangular Current Pulse



Combinational Macro



Dynamic Memory Macro



Static Memory Macro

Figure 2.5 - Examples of three types of macro functions

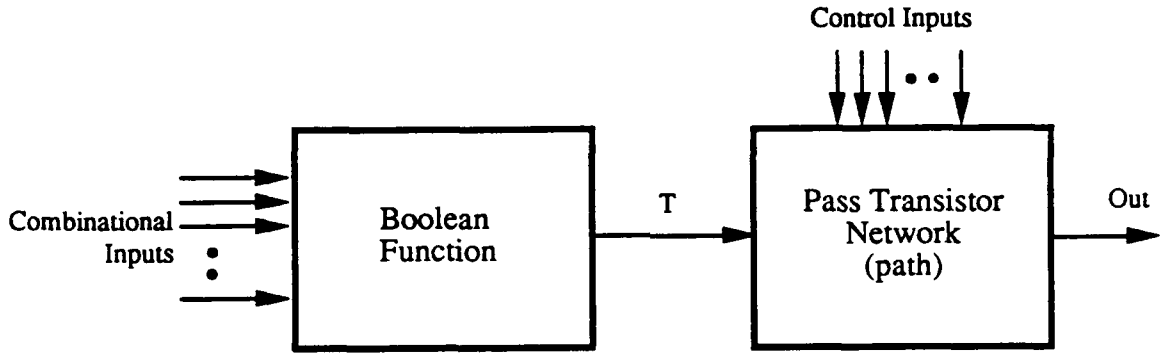


Fig. 2.6 Macro of a dynamic memory consisting of two components

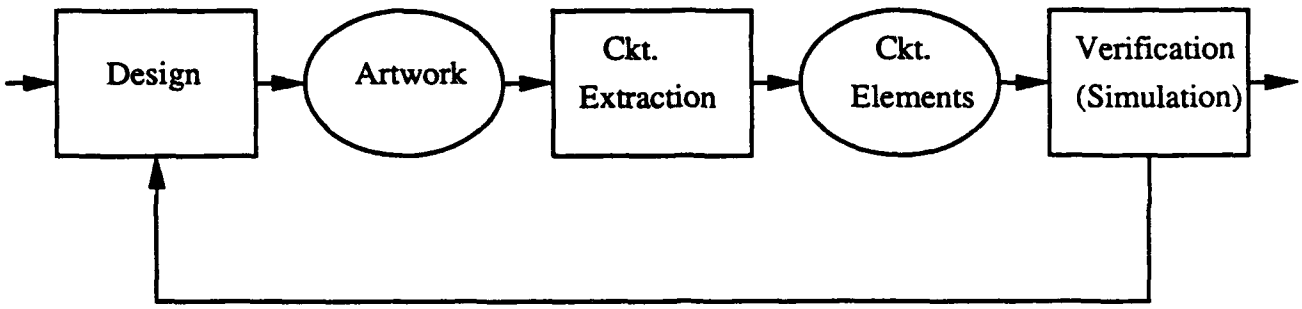


Figure 3.1 - The Artwork Design Process

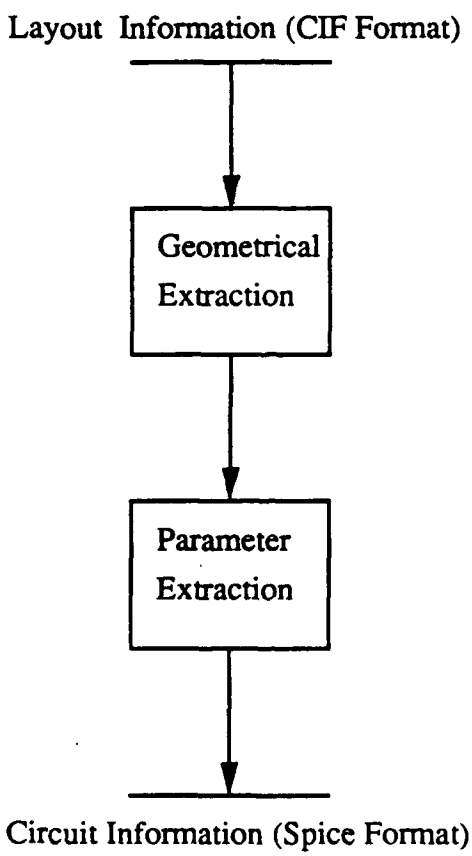


Figure 3.2 - Flowchart for Circuit Extraction

```
ram_cell.cif        Wed Feb  7 03:08:26 1990        1

DS 1;
9 ram_cell;
L CMF;
94 DN 400 1700 CMF;
94 D -500 1700 CMF;
94 Vdd! -2100 4300 CMF;
    B 4500 200 150 4400;
    B 400 800 1500 3900;
    B 400 800 -1100 3900;
    B 400 500 700 3650;
    B 400 1100 -300 3350;
    B 600 500 600 3150;
    B 400 800 500 2500;
    B 500 1900 -250 1850;
    B 200 600 2300 1900;
    B 200 600 -2000 1900;
    B 400 500 1500 1850;
    B 500 1200 550 1500;
    B 400 500 -1200 1850;
    B 1100 400 1850 1400;
    B 1100 400 -1550 1400;
    B 800 200 800 100;
    B 800 200 -500 100;
L CMS;
94 QN 800 0 CMS;
94 Q -1300 0 CMS;
94 GND! -2100 0 CMS;
94 GND! 2200 0 CMS;
    B 200 4500 2300 2250;
    B 400 4300 1400 2350;
    B 400 4300 -1100 2350;
    B 200 4500 -2000 2250;
    B 800 200 1200 100;
    B 800 200 -900 100;
L CPG;
94 row -2100 300 CPG;
    B 200 1500 1100 3350;
    B 200 700 -700 3750;
    B 1600 200 -100 3300;
    B 400 300 500 3050;
    B 200 1200 -800 2600;
    B 400 200 -200 2700;
    B 1600 200 400 2500;
    B 200 1300 1000 1750;
    B 300 200 -750 1900;
    B 200 700 -700 1450;
    B 1300 200 1550 1000;
    B 1300 200 -1250 1000;
    B 4500 200 150 400;
L CAA;
    B 1200 200 100 4400;
    B 1200 400 1100 3700;
    B 1200 400 -700 3700;
    B 800 200 1000 2200;
    B 900 200 -750 2200;
    B 200 400 2300 2000;
    B 200 400 -2000 2000;
    B 1300 700 1050 1750;
    B 1300 700 -750 1750;
    B 1700 800 1150 1000;
    B 1700 800 -850 1000;
    B 1000 600 800 300;
    B 1000 600 -500 300;
L CCA;
    B 200 100 500 4450;
    B 200 100 100 4450;
    B 200 100 -300 4450;
    B 200 200 1500 3700;
    B 200 200 700 3700;
    B 200 200 -300 3700;
    B 200 200 -1100 3700;
    B 100 200 2350 2000;
    B 100 200 -2050 2000;
    B 200 200 1500 1900;
    B 200 200 600 1900;
    B 200 200 -300 1900;
    B 200 200 -1200 1900;
    B 200 200 1500 1500;
    B 200 200 -1200 1500;
    B 200 200 600 1100;
    B 200 200 -300 1100;
    B 200 100 600 50;
    B 200 100 -300 50;
L CWP;
    B 4500 2800 150 1400;
L CVA;
    B 100 200 2350 1400;
    B 100 200 -2050 1400;
    B 200 100 1000 50;
    B 200 100 -700 50;
L COG;
L CCP;
    B 200 200 500 3100;
    B 200 200 -200 2600;
L CSP;
    B 1200 400 1100 3700;
    B 1200 400 -700 3700;
    B 200 400 2300 2000;
    B 200 400 -2000 2000;
DF;

C 1;
E
```

Fig. 3.3a RAM cell CIF input file

```
out          Tue Feb 13 23:29:56 1990          1

* Sizea/Sizeb - 1/1, [Xmin,Ymin,Xmax,Ymax] = [-2100, 0, 2400, 4500]
C0 8 0 5.000e+01pf
* Net 8 row
C0 7 0 1.001e+03pf
* Net 7 D
C0 6 0 9.870e+02pf
* Net 6 DN
C0 5 0 5.020e+02pf
* Net 5 GND!
C0 4 0 2.030e+02pf
* Net 4 Q
C0 3 0 3.418e+02pf
* Net 3 Vdd!
C0 2 0 5.200e+02pf
* Net 2 GND!
C0 1 0 2.030e+02pf
* Net 1 QN
m0_1    3    7    6 1 pdev l=2.00u w=4.00u
m0_2    7    6    3 1 pdev l=2.00u w=4.00u
m0_3    6    7    2 0 ndev l=2.00u w=23.00u
m0_4    7    6    5 0 ndev l=2.00u w=24.00u
m0_5    1    8    6 0 ndev l=2.00u w=10.00u
m0_6    4    8    7 0 ndev l=2.00u w=10.00u
```

Fig. 3.3b RAM cell SPICE file description produced by the extractor

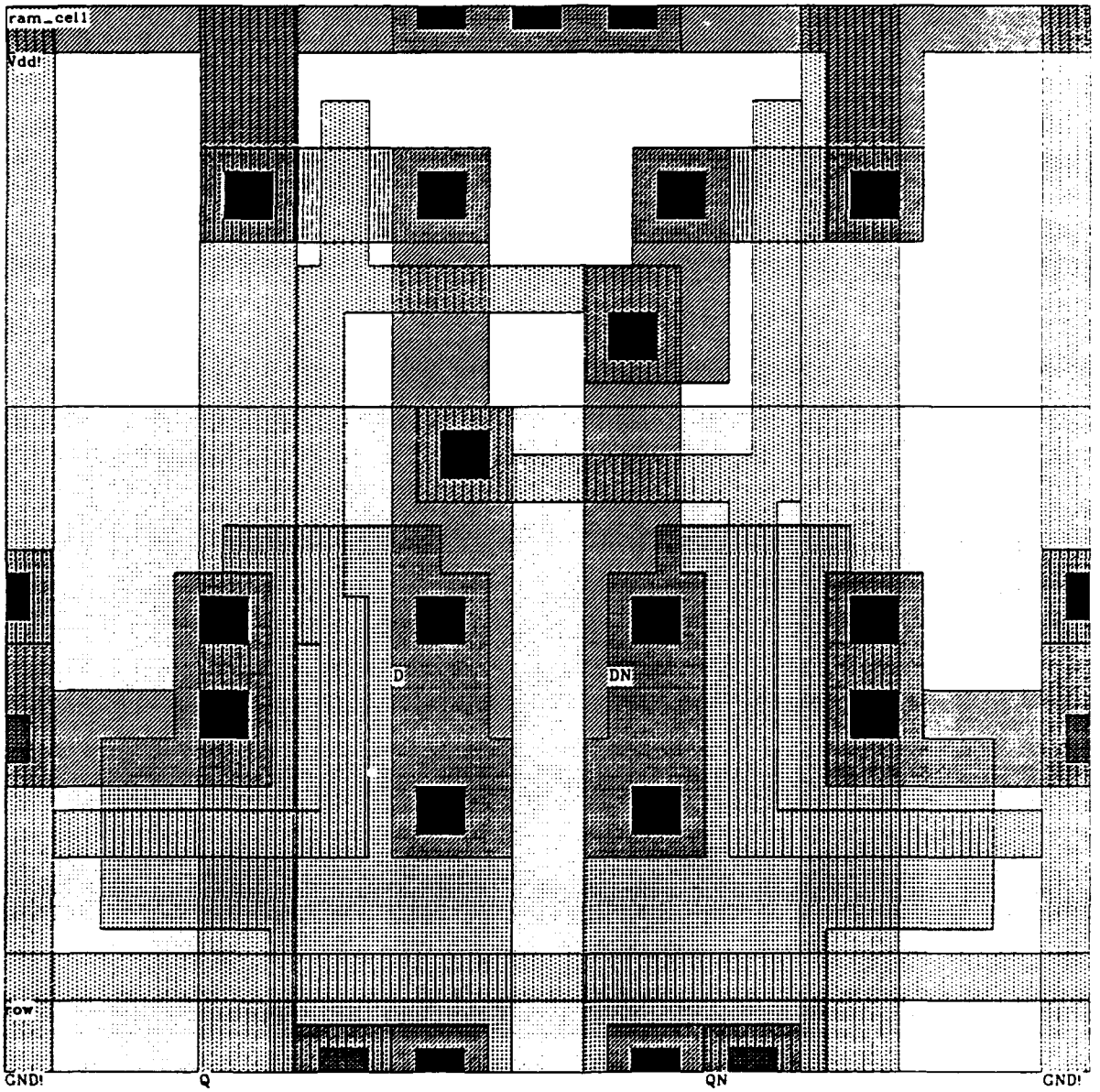


Fig. 3.3c RAM cell layout produced from the Oct/Vem system

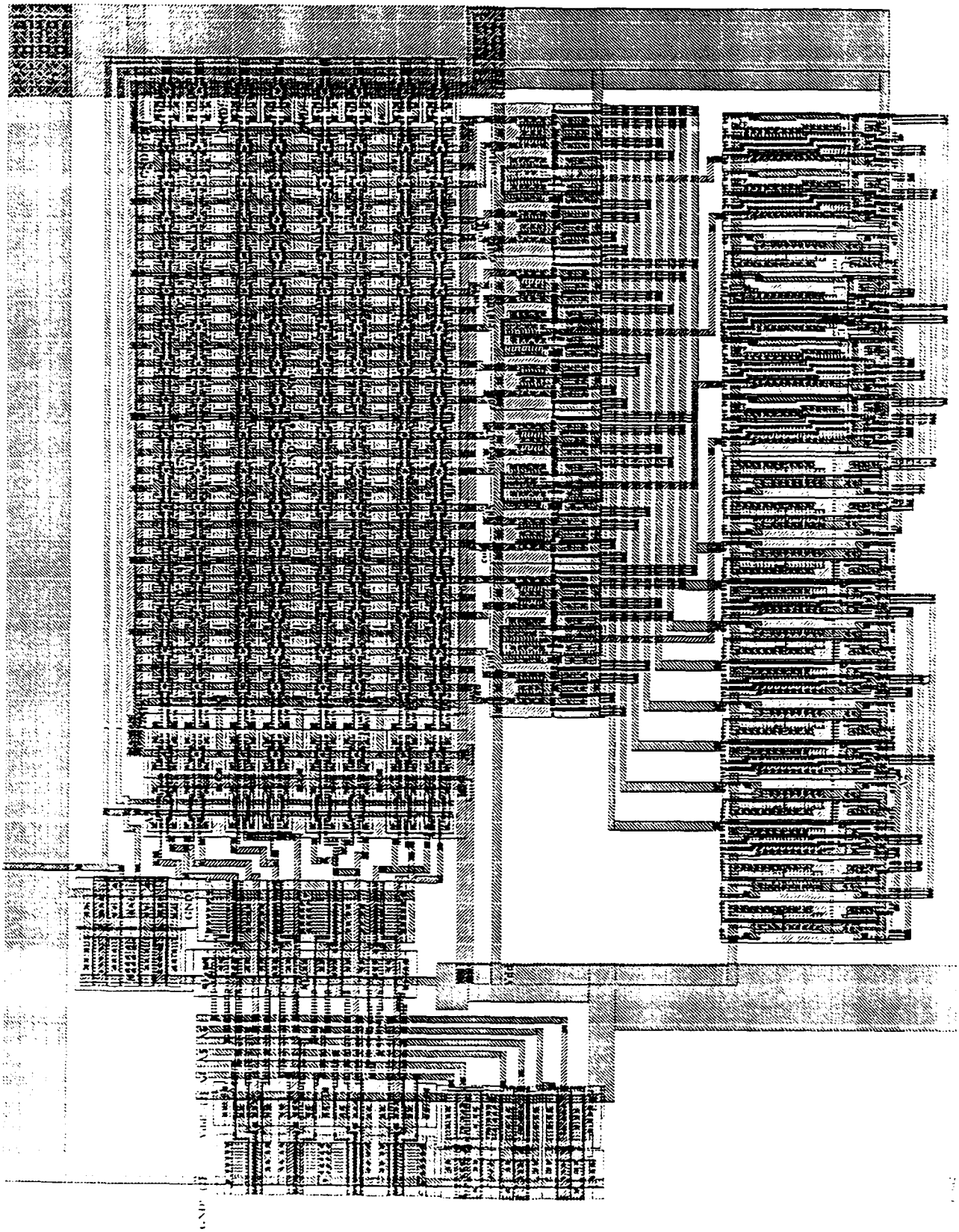


Fig. 3.4a ROM cell layout

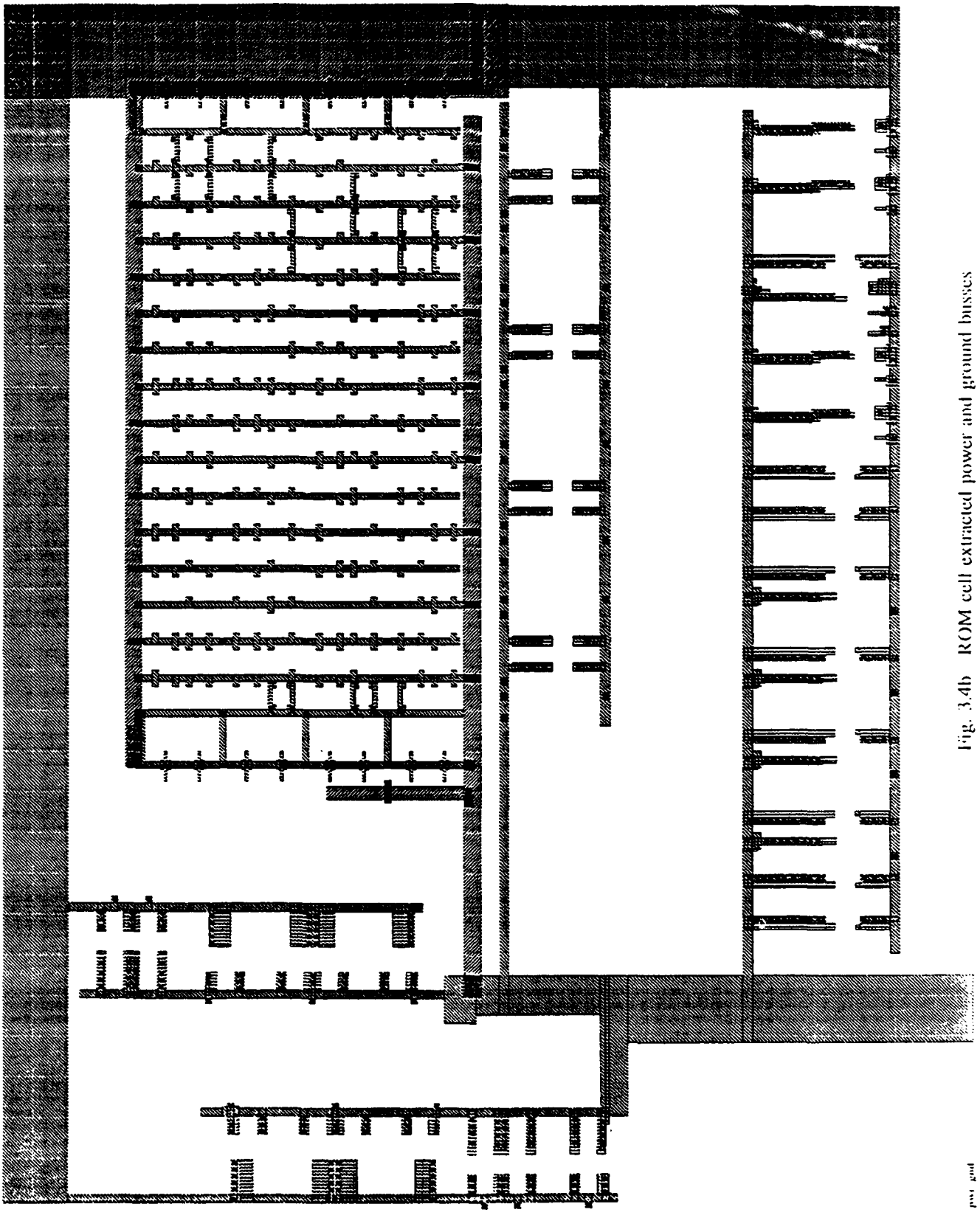


Fig. 3.4b ROM cell extracted power and ground busses

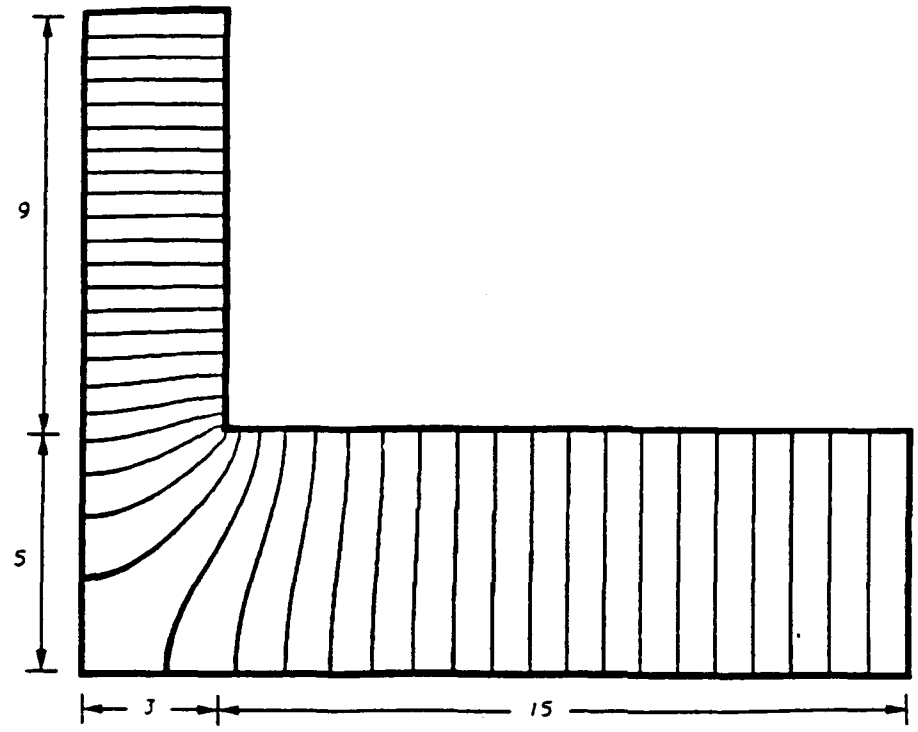


Fig. 4.1 Equipotential lines in a right-angle bend

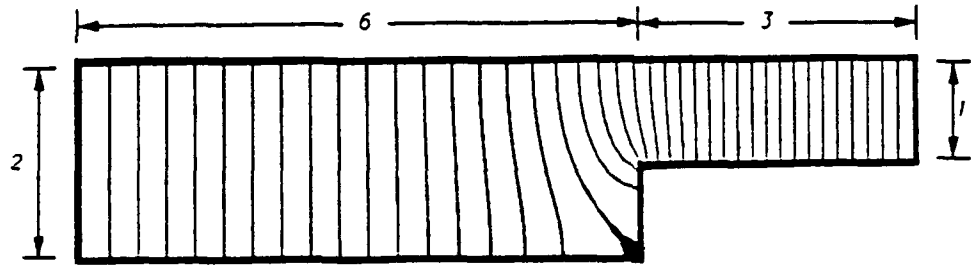


Fig. 4.2 Equipotential lines in a metal region with line width variation

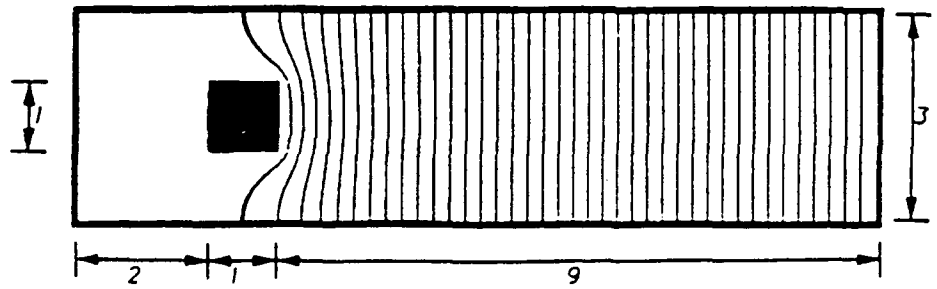


Fig. 4.3 Equipotential lines near contacts

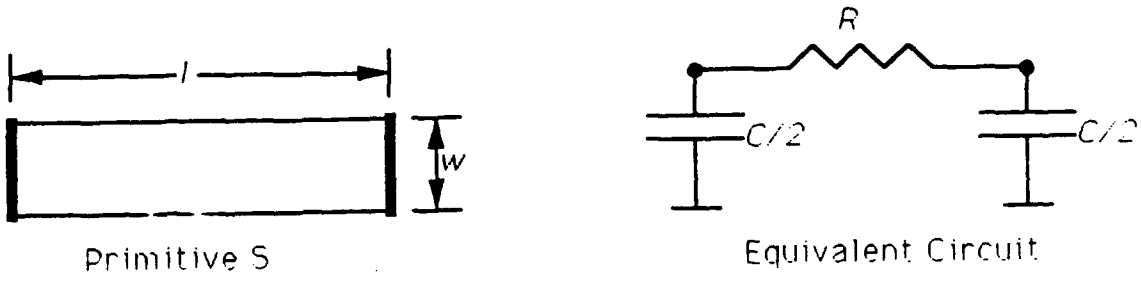


Fig. 4.4 Equivalent RC circuit model of a straight-line bus segment

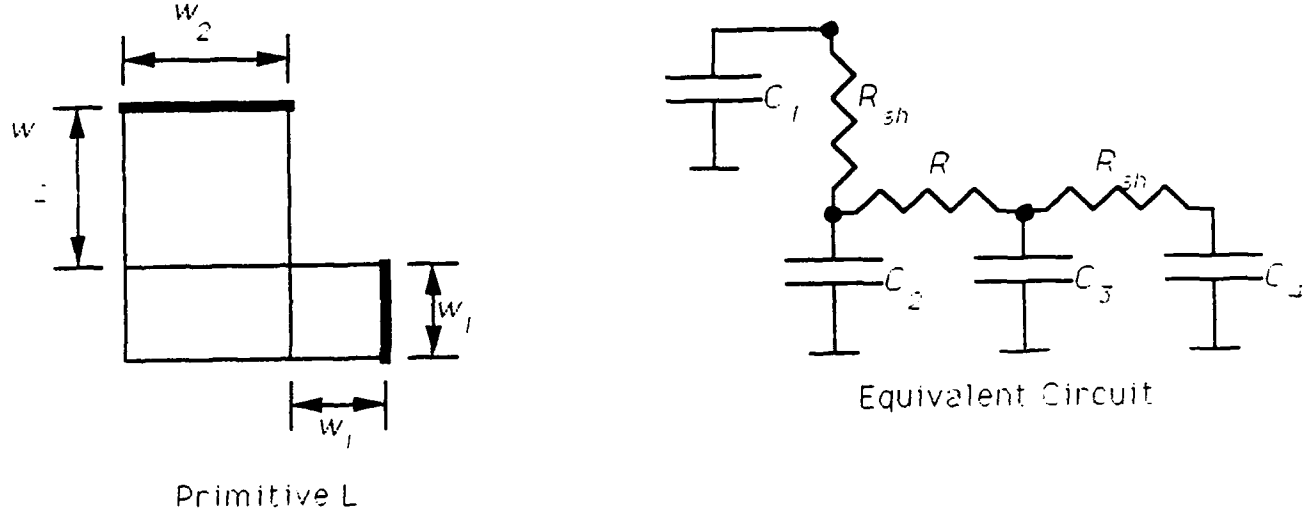


Figure 5

Fig. 4.5 Equivalent RC circuit model of a right-angle bend

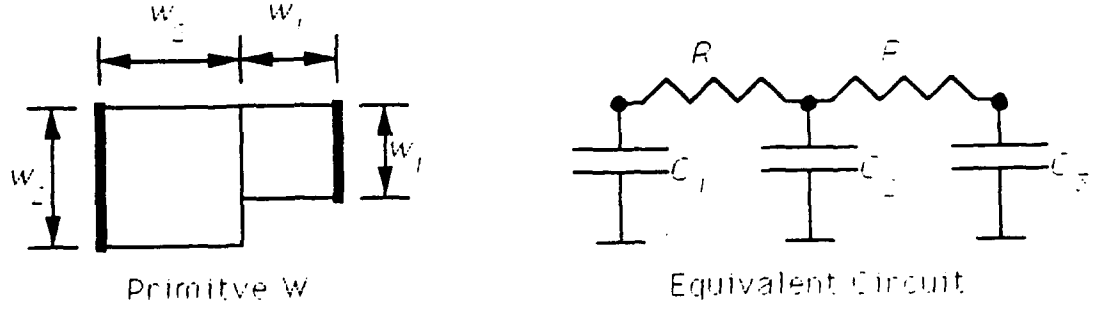


Fig. 4.6 Equivalent RC circuit model of a line width change

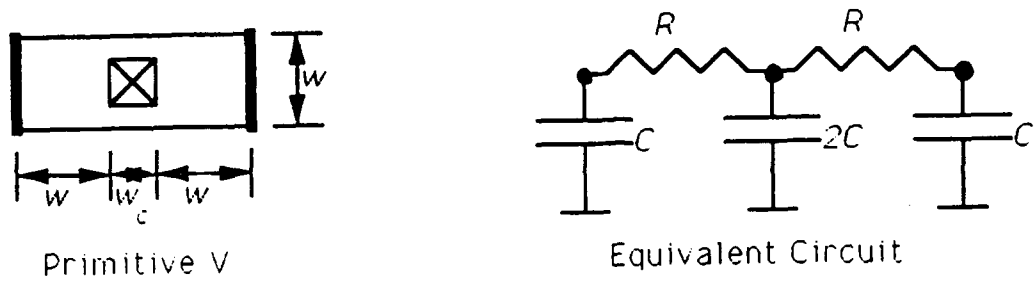


Fig. 4.7 Equivalent RC circuit model near a contact

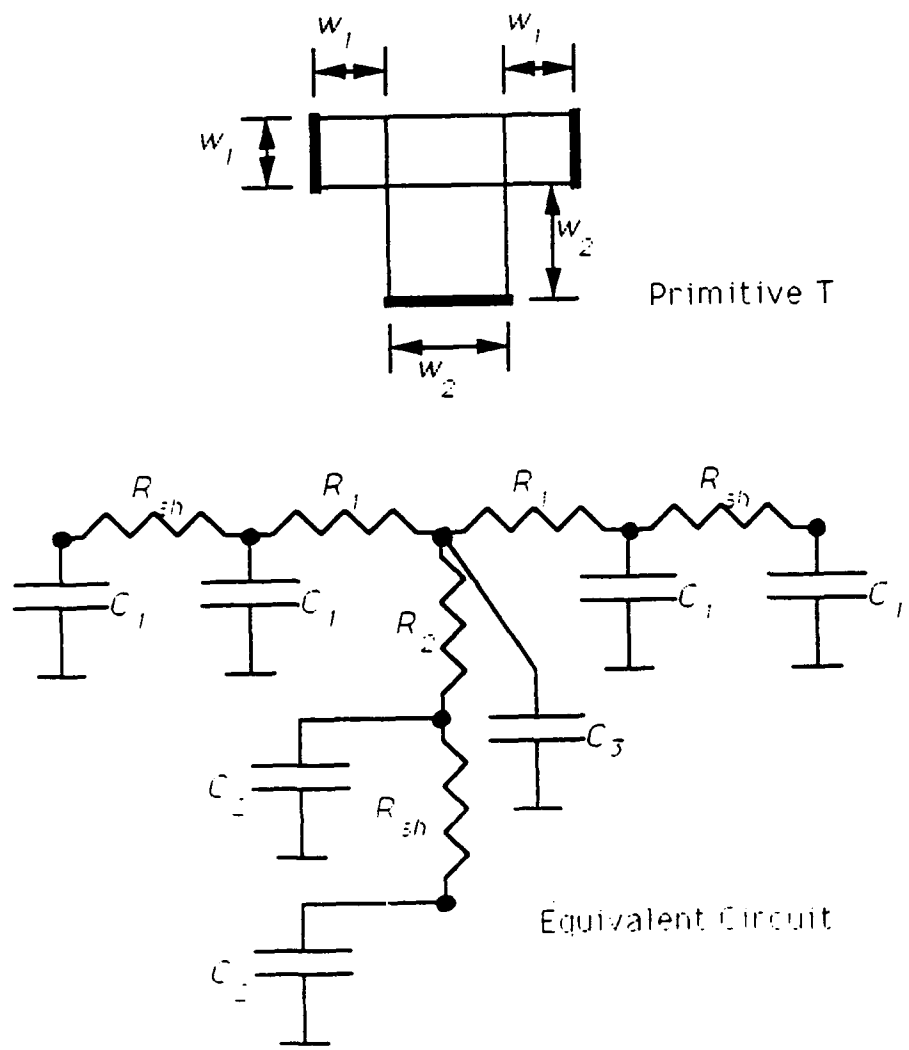


Fig. 4.8 Equivalent RC circuit model of a three-way junction

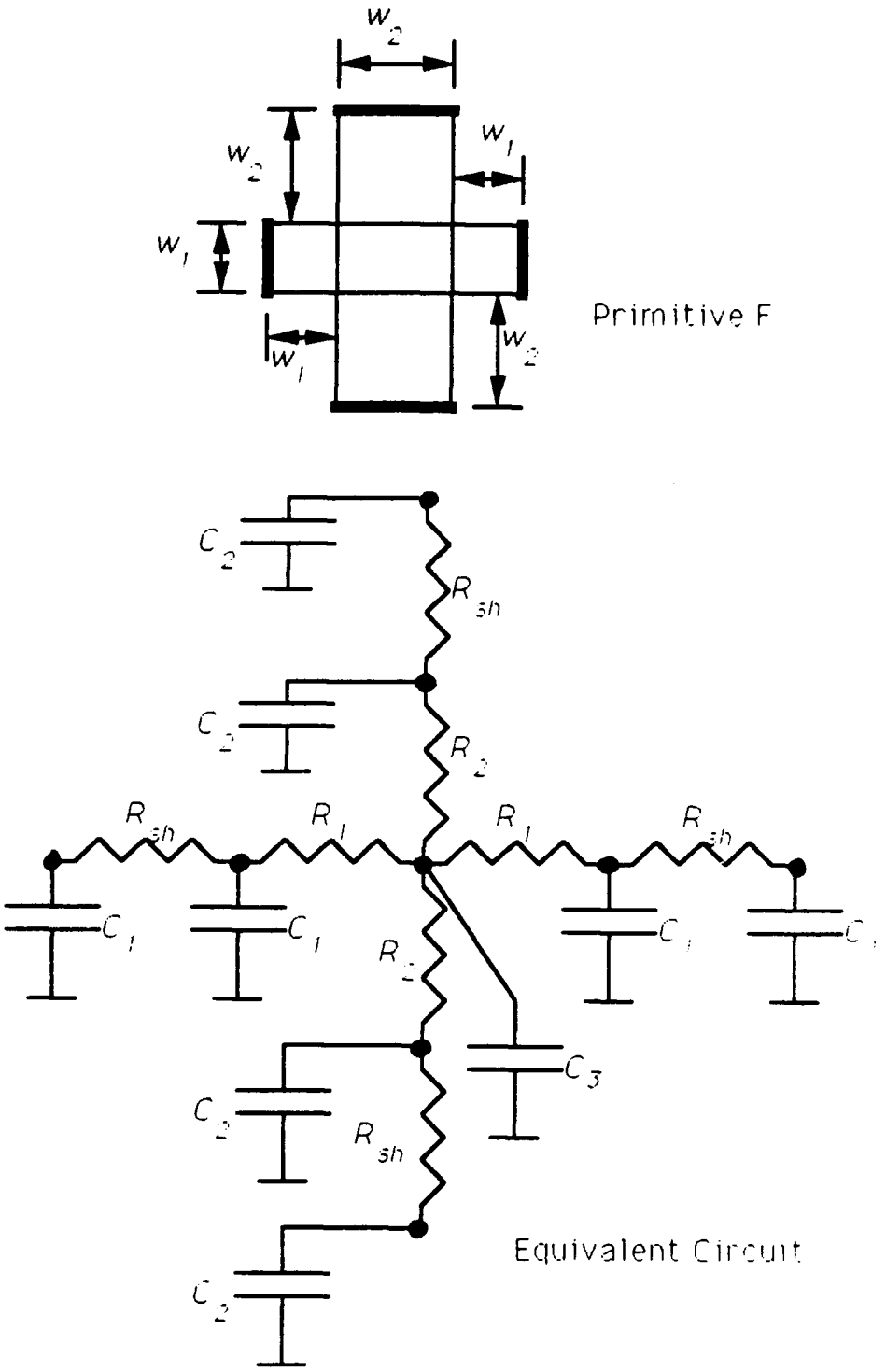


Fig. 4.9 Equivalent RC circuit model of a four-way junction

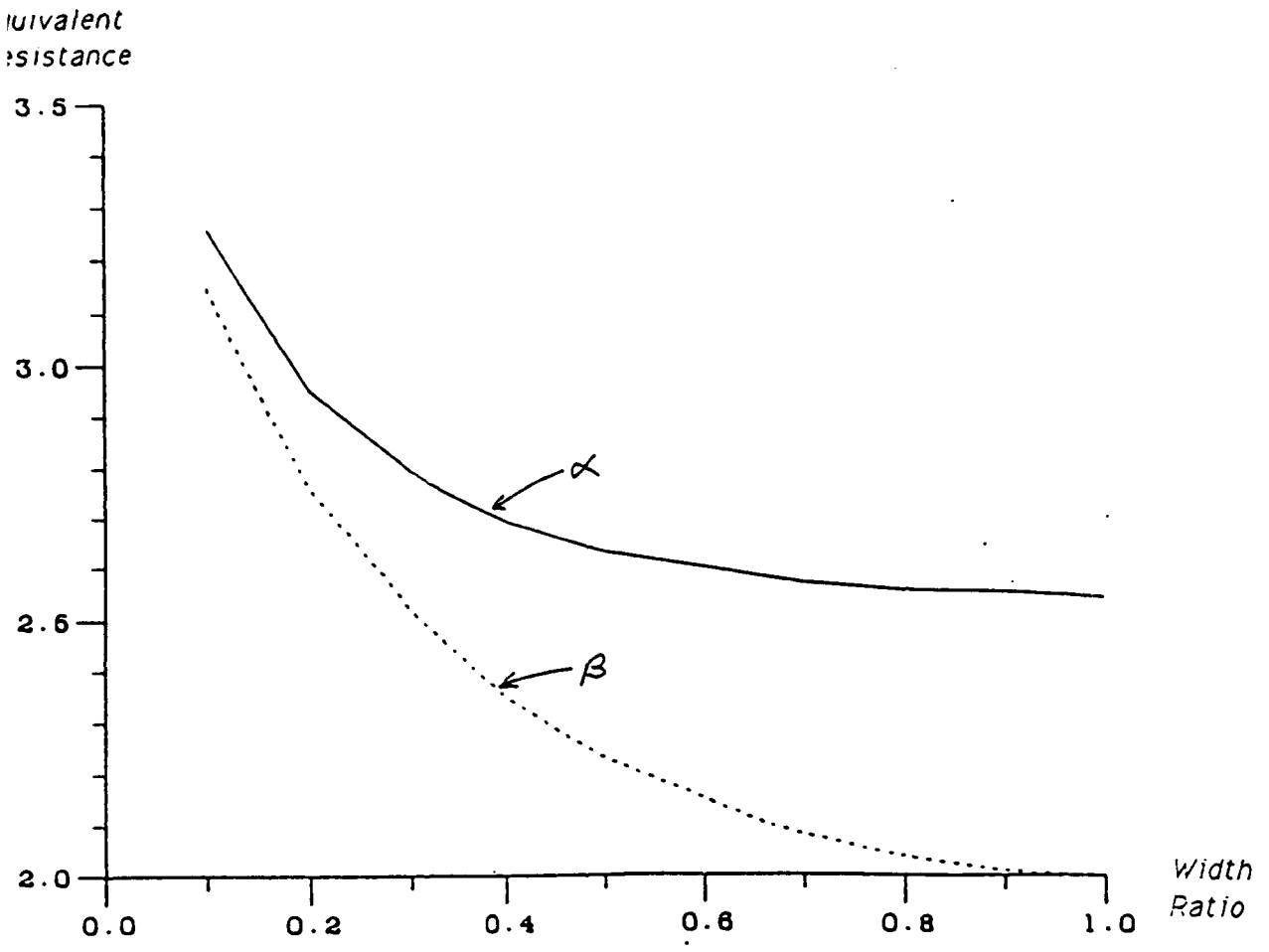


Fig. 4.10 Plots of correction factors α (due to bends) and β (due to line width variation) as functions of line width ratio

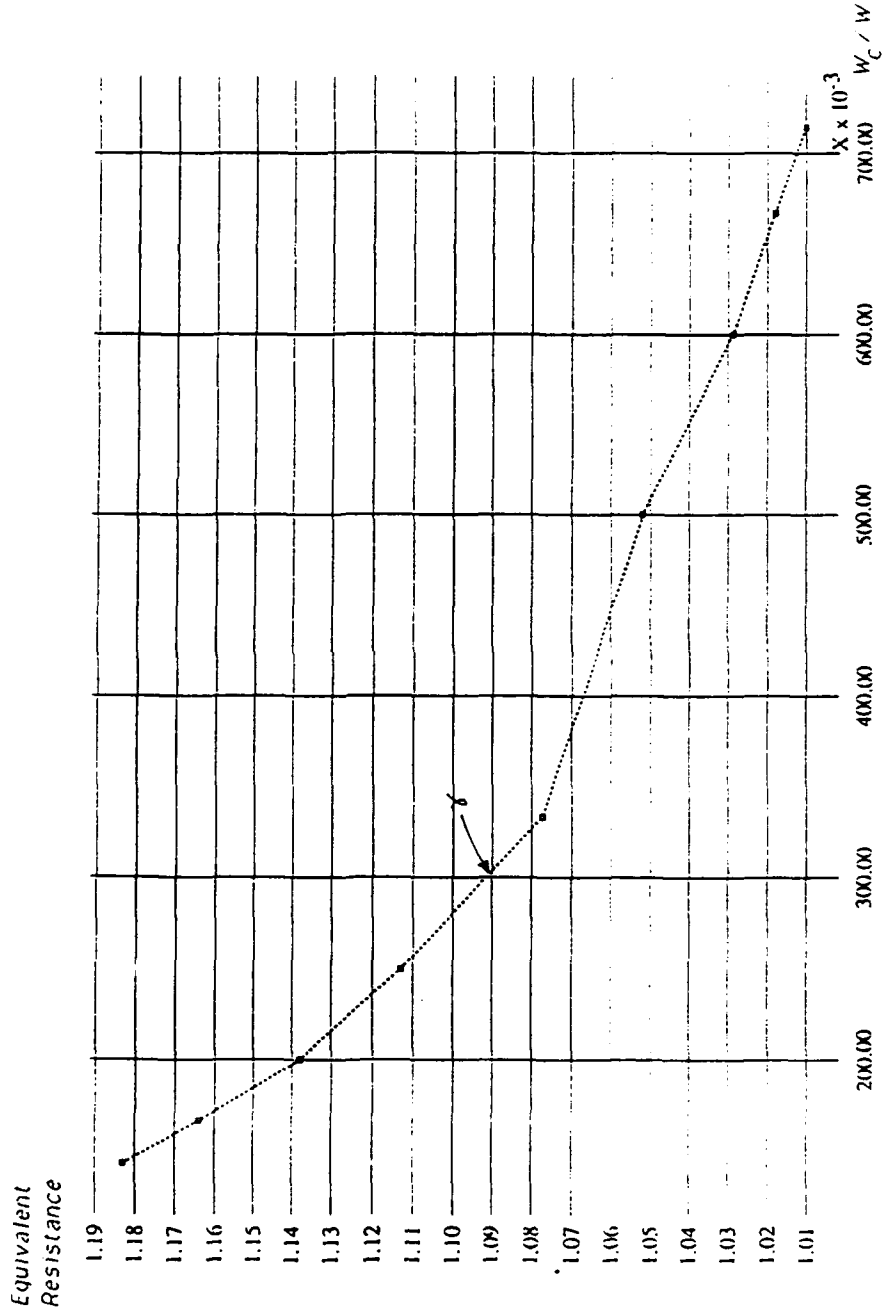


Fig. 4.11 A plot of correction factor γ due to contact region

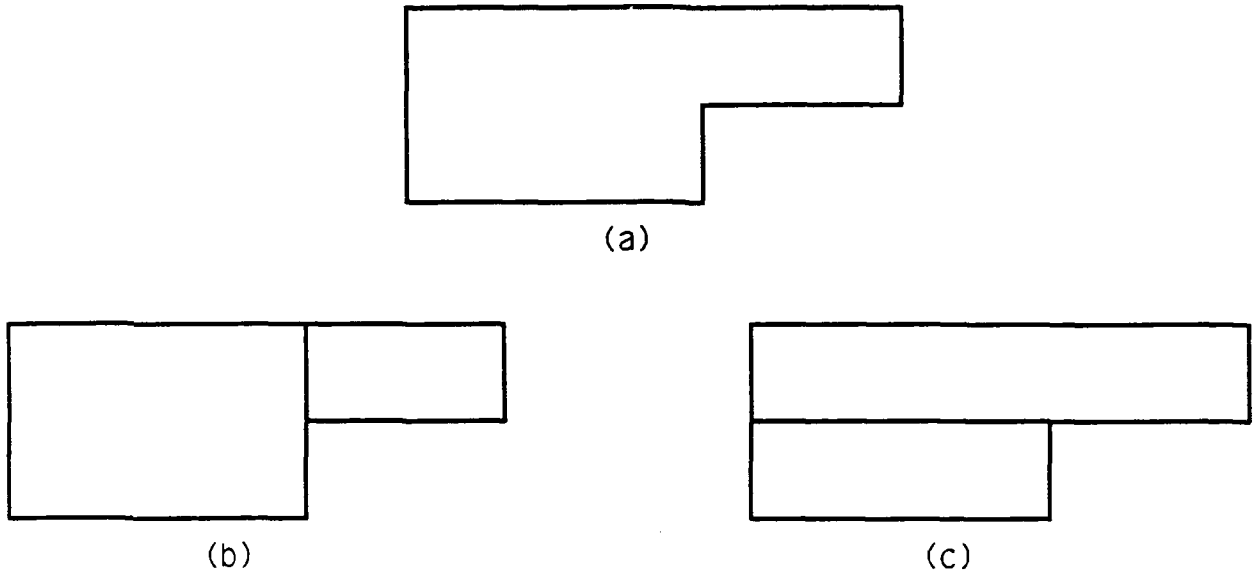


Fig. 4.12 (a) A rectilinear region
(b) Decomposition not in MHS form
(c) Decomposition in MHS form

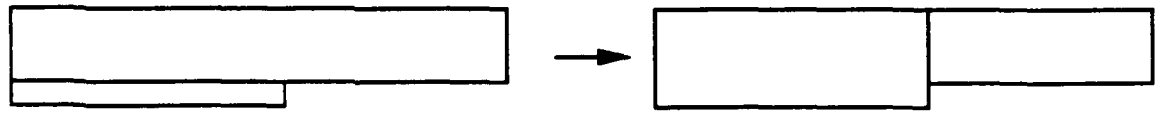


Fig. 4.13 Abutting horizontal rectangles: valid combination

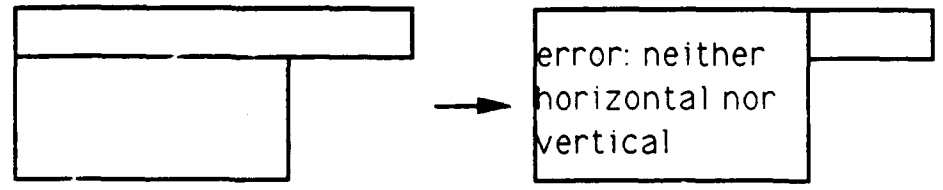


Fig. 4.14 Abutting horizontal rectangles: invalid combination

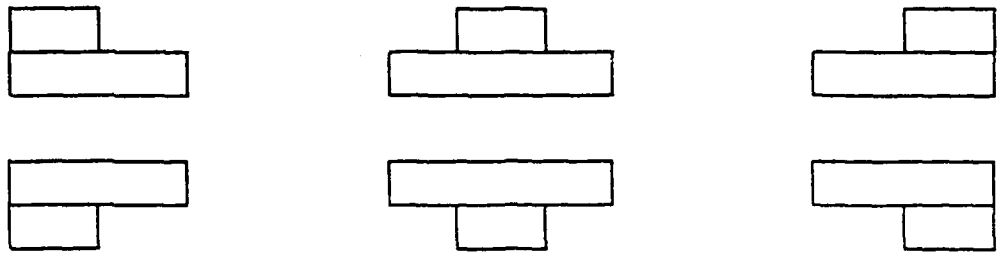


Fig. 4.15 Six different combinations of abutting horizontal rectangles

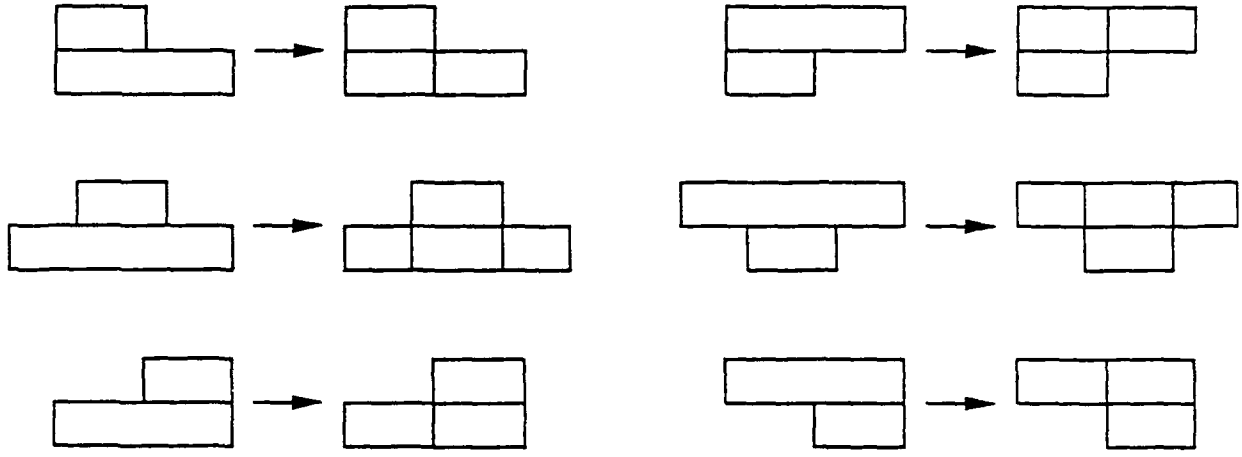


Fig. 4.16 Decomposition of abutting rectangles

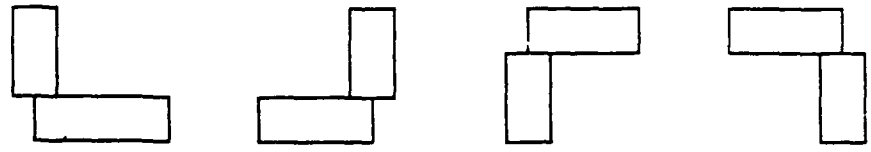


Fig. 4.17 Invalid abuttments of rectangles

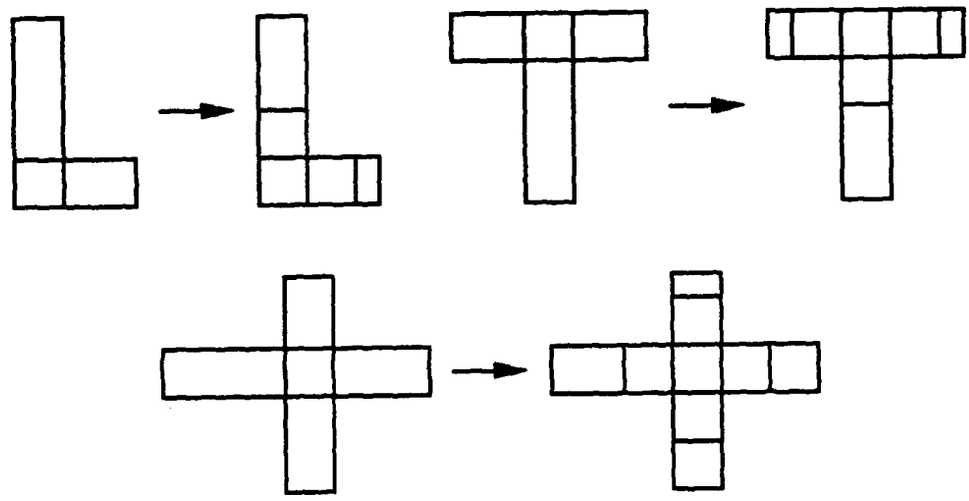


Fig. 4.18 Identification of L, T, and F primitives

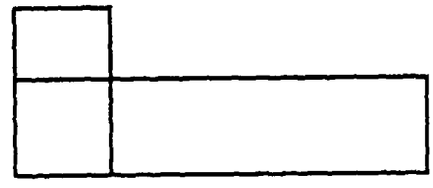


Fig. 4.19 A region which is not an L primitive

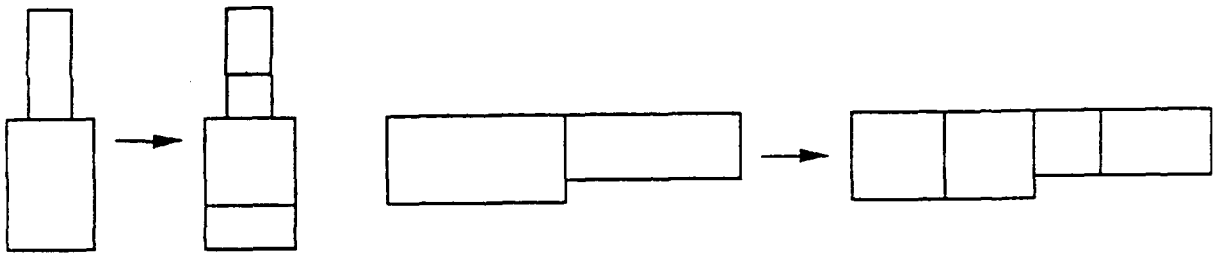


Fig. 4.20 Identification of the W primitive

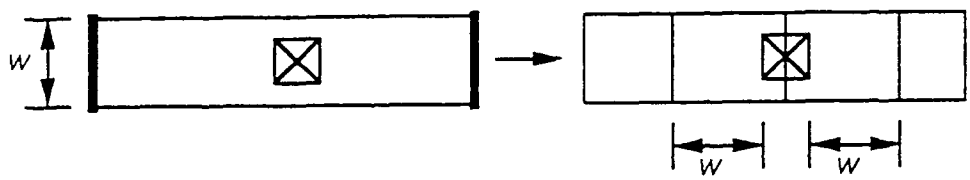


Fig. 4.21 Decomposition of contact region with two sides conducting

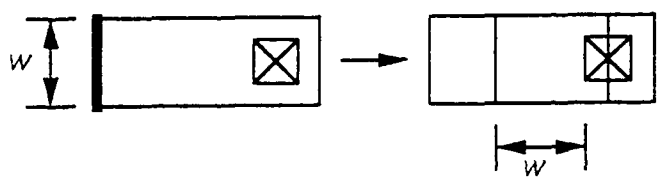


Fig. 4.22 Decomposition of contact region with one side conducting

MISSION

OF

ROME LABORATORY

Rome Laboratory plans and executes an interdisciplinary program in research, development, test, and technology transition in support of Air Force Command, Control, Communications and Intelligence (C^3I) activities for all Air Force platforms. It also executes selected acquisition programs in several areas of expertise. Technical and engineering support within areas of competence is provided to ESD Program Offices (POs) and other ESD elements to perform effective acquisition of C^3I systems. In addition, Rome Laboratory's technology supports other AFSC Product Divisions, the Air Force user community, and other DOD and non-DOD agencies. Rome Laboratory maintains technical competence and research programs in areas including, but not limited to, communications, command and control, battle management, intelligence information processing, computational sciences and software producibility, wide area surveillance/sensors, signal processing, solid state sciences, photonics, electromagnetic technology, superconductivity, and electronic reliability/maintainability and testability.